Small Sample Size Learning in Bioinformatics

Justin Bedo

April 2009

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

To my parents

Declaration

The work in this thesis is my own except where otherwise stated.

Justin Bedo

Acknowledgements

The past few years has brought me into contact with many people from whom I benefited greatly. I am very grateful to Karsten Borgwardt, Arthur Gretton, and Le Song, with whom I collaborated on the Hilbert-Schmidt independence criterion for supervised feature selection. I also benefited greatly from discussions with Simon Günter and Brian Parker, with whom investigations into stratification bias were undertaken.

Additionally, I have benefited greatly from interactions with the Diversity Arrays Technology team in Canberra. Amongst them, I specifically thank Eric Huttner, Grzegorz Uszynski, and Peter Wenzl, all of whom I've had detailed discussions with.

I would like to thank Conrad Sanderson, with whom the initial work on centroid classifiers was conducted.

I especially thank my advisors Andrzej Kilian, Adam Kowalczyk, and Alex Smola for support, criticism, and advice. In particular, Adam has provided considerable support, for which I'm indebted.

Many thanks to Justin Zobel for support provided in Melbourne.

Finally, a special thanks to my friends Bryan Beresford-Smith, Yao-ban Chan, Alex Chapman, Peter Tomlins, and Gerard Wong for reading drafts of this thesis.

Abstract

The field of bioinformatics is very diverse providing many data mining opportunities. Many applications in genomics involve predicting a target from genetic data obtained using modern technologies such as Microarrays. These technologies have very high resolution, allowing many measurements to be taken simultaneously (well over a million measurements per sample are currently possible). This high-dimensionality is typically not accompanied by a large sample size; many studies consist of a few hundred or fewer samples, and frequently fewer than one hundred samples. Learning from such data sets is not always straight forward, and strange learning phenomena can arise. This thesis studies several distinct feature selection problems in bioinformatics and investigates the strange small sample learning phenomenon of antilearning.

Two novel methods for quantitative trait loci (QTL) mapping are presented and evaluated on both synthetic and natural data. These methods approach the problem of QTL mapping using generalisation estimation rather than using null-hypothesis testing like many traditional approaches.

Several theoretical links between the support vector machine (SVM) classifier and a centroid classifier are presented, showing that the SVM converges to a centroid classifier in the limit of high regularisation. These theoretical links were then used to derive a feature filter and centroid classifier combination, which is the limit of the *recursive feature elimination* SVM in the limit of high regularisation. The centroid combination was evaluated on several cancer datasets and shown to perform well in comparison to other methods, despite its inherent simplicity.

A novel method for unsupervised feature selection is also presented and studied, inspired by the problem of microarray design for sugarcane crops. This unsupervised method is shown to perform well in comparison to supervised methods.

Finally, the strange small-sample learning phenomenon called antilearning is explored and a method for detecting the mode of the data is presented. Using this method, a *reversible learner* is explored which detects and corrects for anti-learnable data. This reversible learner is shown to generalise correctly on both anti-learnable and learnable data.

Contents

Chapter 1

Introduction

Technologies such as microarrays (McLachlan et al., 2004; Speed, 2003) and high-throughput sequencing have resulted in a rapid growth in the level of detail that was previously unreachable due to technical or cost limitations; microarrays allow the measurement of gene expression levels for tens of thousands of genes, and single nucleotide polymorphism (SNP) microarrays can detect genomic differences at half a million SNPs or more. Unfortunately, this large growth in the resolution of data was not accompanied by a growth in the size of studies. As an example, studies in cancer genomics typically consist of less than 100 patients due to availability restrictions.

This thesis focuses on learning from small sample bioinformatics datasets. Various data mining techniques are explored in an effort to extract usable information from the data. In particular, this thesis is concerned with *feature selection* for knowledge discovery and improved performance. Herein, feature selection is used for detection of relevant genes to guide further biological research – for example in plant breeding and cancer genomics – and also to allow lower resolution technologies to be used for more cost effective clinical tests.

Like bioinformatics itself, this thesis is very diverse. Plant genomics, cancer genomics, machine learning, and topology are all touched on. The applications studied herein are varied, and range from survival prediction for cancer patients to microarray design for arraying of sugarcane crops. A strange phenomenon that can arise in the area of small sample learning, namely the phenomenon of *antilearning*, is also studied.

1.1 Overview

Chapter 2 introduces concepts from statistical machine learning that are used throughout this thesis. In particular, the concepts of probability, empirical risk minimisation, feature selection, reproducing kernel Hilbert spaces, and linear learning machines, such as support vector machines and ridge regression, are reviewed and presented.

Chapter 3 reviews classical *quantitative trait loci* (QTL) mapping techniques and introduces two novel approaches towards mapping QTL in plant genomics. Many previous QTL methods have followed a traditional statistical approach whereby a model is built by considering putative QTL locations independently and using null-hypothesis testing. The new methods presented in this chapter analyse the whole genome simultaneously, with a strong focus on *generalisation ability* rather than hypothesis testing. The two novel methods – one based on obtaining sparsity through *recursive feature elimination* (RFE), and the other through regularisation – are benchmarked against currently used QTL mapping methods on both synthetic and natural data. The new methods were shown to perform well in comparison. The RFE method has been published (Bedo et al., 2008), but the second is currently unpublished.

Unlike the regression problem studied in the QTL mapping chapter, Chapter 4 focuses on small sample *classification* problems. Here, the limit of high-regularisation support vector machines (SVM) is shown to converge to a centroid classifier. Furthermore, the limit of non-linear performance metrics is shown to converge when certain criteria are satisfied. A direct consequence of this result is that the recursive feature elimination SVM (RFE–SVM) converges to a simple centroid classifier in combination with a feature filter. This *centroid method* is shown to perform well on several cancer genomics datasets, including a multiclass dataset with numerous classes with sparse representation. The chapter extends work originally presented by Bedo et al. (2006).

Chapter 5 considers the task of microarray design for arraying of sugarcane data. The inspiring problem behind this dataset is to select a 6912 feature subset from an initial 50,000 feature set. This is an unsupervised feature selection task, and a method based on the *Hilbert-Schmidt independence criterion* (HSIC), named *unsupervised feature selection by the Hilbert–Schmidt independence criterion* (UBHSIC, pronounced ['u.bə-sik]) is presented. The performance of UBHSIC was evaluated by comparing the performance on the HSIC reduced dataset and the full dataset for several cancer genomics datasets. This method is an unsupervised

variant of the supervised HSIC based feature selector previously published by Song et al. (2007b,a), but uses a quantum annealing optimiser instead of nested subset selection.

Finally, Chapter 6 discusses the strange phenomenon of *antilearning* that can occur in small sample learning. Antilearning is characterised by good performance on the training set, but *below random* performance on the independent test set. In the case of two-class classification, the classifier consistently predicts future samples incorrectly and an accurate predictor can be obtained by simply reversing the predictions. The phenomenon is introduced using synthetic data, and also shown to arise naturally in a cancer genomics dataset. Feature selection is shown not to achieve standard learning behaviour on the antilearnable datasets studied. This chapter is based on the paper by Kowalczyk et al. (2007), but extends prior work by presenting a new kernel based method for the detection and subsequent inversion of predictions in the case of antilearning data.

Chapter 2

Statistical Machine Learning

This chapter introduces statistical machine learning (SML) concepts that are used throughout this thesis. To begin, the core terminology and notation used in the sections to come are introduced.

Throughout this thesis, $\mathbb{R}$ denotes the set of real numbers and $\mathbb{R}^n$ the set of n-tuples

$$\mathbf{x} := (x_1, x_2, \ldots, x_n)$$

where $x_i \in \mathbb{R}$. All n-tuples are indicated by a bold font, and all elements of $\mathbb{R}$ by a lowercase font. The standard dot product between two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ is denoted by

$$\langle \mathbf{x}, \mathbf{x}' \rangle := x_1 x_1' + x_2 x_2' + \cdots + x_n x_n'.$$

Capital letters indicate $n \times m$ matrices of the form

$$X := \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

where $\mathbf{x}_i$ denotes the i^{th} row and $\mathbf{x}^{(j)}$ denotes the j^{th} column of matrix X. The transpose of a matrix is denoted X^*. Vectors are always column vectors.

The notation $|\cdot|$ denotes both set cardinality when applied to a set, and absolute value when applied to a number. The notation

$$||\mathbf{x}||_p := (\sum_i |x_i^p|)^{\frac{1}{p}}$$

denotes the L^p-norm of $\mathbf{x}$ for $p \in \{1, 2, \dots\}$. Furthermore, the L^0-norm and L^∞-norm are defined as

$$||\mathbf{x}||_0 := \sum_i 1 - \delta(x_i) \text{ and } ||x||_\infty := \sup_i |x_i|,$$

where

$$\delta(x) \colon \mathbb{R} \to \{0, 1\}$$
$$x \mapsto \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

is the Dirac delta function.

For classification and regression learning problems, the available data samples are denoted

$$\mathcal{X} := \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{X} \times \mathbb{Y} \subset \mathbb{R}^m \times \mathbb{R} \tag{2.1}$$

where $\mathbb{X} \subset \mathbb{R}^m$ is the space of measurement vectors and $\mathbb{Y} \subset \mathbb{R}$ is the set of labels. For regression problems $\mathbb{Y}$ is assumed to be the entire real line $\mathbb{R}$, and for classification $\mathbb{Y}$ is a finite set. In the simple case of two-class classification, it is assumed $\mathbb{Y} = \{1, -1\}$.

Finally, $\mathcal{F}$ always denotes a set of functions, which is called the *hypothesis space*

$$\mathcal{F} \subset \{f | f \colon \mathbb{X} \to \mathbb{Y}\} =: \mathbb{Y}^{\mathbb{X}}.$$

2.1 Probability Measures and Expected Values

To begin, it is necessary to define some concepts relating to probability, namely the concepts of *probability measures* and the *expected value* of a function. The treatment here is short; more information regarding measure theory can be found elsewhere (Dudley, 1987).

A probability measure assigns some non-negative probability to an event occurring. A simple example is the toss of a fair coin where the two possible events are either heads H or tails T. If $P(\dots)$ is used to represent "the probability of ...", then $P(H)$ and $P(T)$ can be assigned $P(H) = P(T) = 0.5$, and thus $P(H) + P(T) = P(H \cup T) = 1$. Furthermore, as it was specified that one toss has

been carried out, $P(\emptyset) = 0$. Let us now examine the properties that were used in defining the probability function P. First, $P(\emptyset) = 0$ and $P(H \cup T) = 1$, that is the probability of nothing occurring is 0 and the probability of anything occurring is 1. Second, the additivity of disjoint events, that is $P(H) + P(T) = P(H \cup T)$, was used. This framework can be extended easily to any finite space of outcomes.

As the number of coin flips increases, the probability of any specific sequence of outcomes decreases. If the number of flips goes to infinity, then any specific sequence has probability 0. This, however, does not exclude the possibility of interesting events occurring. For example, the probability of the first $n < \infty$ flips having a specific sequence is non-zero. What is desired is the properties outlined previously extended to an infinite space of continuous outcomes. Such concepts are provided by Lebesgue and Borel measures from measure theory (Dudley, 1987).

For continuous spaces such as $\mathbb{R}^m$, the events must be restricted to certain subsets to avoid contradictions. This set of subsets is called a σ-algebra. Any σ-algebra must have the following properties:

1. the σ-algebra is non-empty;
2. the complement of any set in the σ-algebra is contained in the σ-algebra;
3. any countable union of sets contained in the σ-algebra is also contained in the σ-algebra.

The Borel σ-algebra of $\mathbb{R}^m$ is the smallest σ-algebra containing the open subsets[1].

Definition 2.1 (Borel probability measure (Dudley, 1987)). *Let $\mathcal{A}$ be the Borel σ-algebra of $\mathbb{X}$. A function $P\colon \mathcal{A} \to \mathbb{R}^+$ is a Borel probability measure with $P(\emptyset) = 0$ and $P(\mathbb{X}) = 1$ if it is σ-additive, i.e., $P(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} P(A_i)$ for any disjoint subsets $A_1, A_2, \ldots, A_n \in \mathcal{A}$.*

Given an event and a Borel probability measure, the probability can be determined if the event falls within the σ-algebra. For example, the probability of a function $f\colon \mathbb{R}^n \to \mathbb{R}$ being greater than zero given a measure P is $P(\{x|f(x) > 0\})$, or written as simply $P(f(x) > 0)$, under the condition that the set $\{x|f(x) > 0\}$ is *measurable* (i.e., $\{x|f(x) > 0\}$ belongs to the σ-algebra).

[1] This can be generalised to topological spaces.

This restricts us to the class of *measurable functions*, which includes all *continuous functions*.

Let us consider two sets in the σ-algebra, A and B, and assume $P(B) > 0$. The *conditional probability* of A given B is

$$P(A|B) := P(A, B)/P(B)$$

where $P(A, B) := P(A \cap B)$, and is known as *Bayes' rule*.

Using this probability measure and Lebesgue integration (Dudley, 1987), the *expected value* of a function can be defined.

Definition 2.2 (Expected value (Dudley, 1987)). *The expected value of a (measurable) function f given a probability measure P is*

$$E_{\mathbf{x}\sim P}[f(\mathbf{x})] := \int f(\mathbf{x})P(d\mathbf{x})$$

where the integral is the Lebesgue integral. The abbreviated notation

$$E_P[f] = \int f dP := \int f(\mathbf{x})P(d\mathbf{x})$$

is used when confusion cannot arise. Furthermore, the notation $E[f]$ is used when P is assumed to be the underlying probability distribution generating the dataset at hand.

Note that f being bounded is a sufficient condition for existence of the expectation.

In the case where P is unknown but a finite sample of *independent and identically distributed* (IID) data from the distribution P is available, an empirical estimate of the expectation can be obtained by calculating the mean. As an example, for a finite sample of data $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{X} \times \mathbb{Y}$ and $f\colon \mathbb{X} \to \mathbb{R}$, the *empirical expected value* of f is

$$E_{\mathcal{X}}^{\text{emp}}[f] := \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{x}_i).$$

In the limit of infinite IID samples, this empirical estimate converges to the actual expectation *almost surely* (i.e., with probability 1), as specified by the *Glivenko-Cantelli* theorem (Talagrand, 1987) and its extensions (e.g., Fortet and Mourier (1953)).

Definition 2.3 (Conditional expected value). *The conditional expected value of*

a (measurable) function f given a probability measure P and an event B, where $P(B) > 0$, is

$$E_{\mathbf{x}\sim P}[f(\mathbf{x})|B] := \int_B f(\mathbf{x})P(d\mathbf{x})/P(B)$$

where the integral is the Lebesgue integral. As before, the abbreviated notations

$$E_P[f|B] = \int_B f dP/P(B) := \int_B f(\mathbf{x})P(d\mathbf{x})/P(B)$$

and $E[f|B]$ are used when confusion cannot arise.

Again, the conditional expectation may not be well-defined, but f bounded and $P(B) > 0$ are sufficient conditions.

2.2 Goodness of Fit

The goal of learning is to find a function $f\colon \mathbb{X} \to \mathbb{Y}$ that is able to estimate the label $y \in \mathbb{Y}$ accurately from a measurement vector $\mathbf{x} \in \mathbb{X}$. Before delving into finding a suitable $f \in \mathcal{F}$, the concept of "accuracy" is explored by defining the metrics for classification and regression; these metrics determine the performance of an hypothesis $f \in \mathcal{F}$. We make a distinction between the goodness-of-fit (GOF) measures presented here, which are intended for easy interpretation, and the loss functions designed for optimisation introduced in the latter sections.

The GOF measures differ between regression and classification problems. As 2-class classification is somewhat similar to regression, the regression measures can be used to evaluate the degree of fit. However, the classification measures are more easily interpretable, and hence are a better choice for performance evaluation.

2.2.1 Regression

One of the simplest measures for regression problems is the *residual sum of squares* (RSS, Hastie et al. 2001). The RSS is not easily interpretable as it is an absolute measure not a relative measure, hence it is more useful for optimisation and model comparison rather than obtaining an intuitive indication of the predictive performance. However, it does form the basis of the much more interpretable *variance explained* measure, which will be introduced directly after the RSS.

Definition 2.4 (Residual sum of squares). *The Residual Sum of Squares* (RSS) *is*

$$\text{RSS}(\mathcal{X}, f) := \sum_{(\mathbf{x},y)\in\mathcal{X}} (y - f(\mathbf{x}))^2$$

where $\mathcal{X}$ is as in Equation 2.1.

Models with low RSS fit the data better than models with a higher RSS. An RSS of 0 indicates the model fits the data perfectly. Unfortunately, as the range of the RSS is dependent on the range of the target y, it can only be used for ordering the performance of various hypotheses $f \in \mathcal{F}$. The *variance explained* measure is a modification of the RSS to normalise the range and provide a relative measure of performance.

Definition 2.5 (Variance explained). *The variance explained is*

$$r^2(\mathcal{X}, f) := 1 - \frac{\text{RSS}(\mathcal{X}, f)}{\sum_{(\mathbf{x},y)\in\mathcal{X}} \left(y - \frac{1}{|\mathcal{X}|}\sum_{(\cdot,y')\in\mathcal{X}} y'\right)^2}$$

The variance explained measure calculates the proportion of variance explained by the model compared against the total variance present in the target y. The value 1 indicates that the model explains 100% of the variance, 0 indicates that no variance was explained, and less than 0 indicates that the model is *adding* variance.

2.2.2 Classification

There are three main measures for evaluating the performance of classification hypothesis used herein: the *error rate*, *balanced error rate*, and the *area under the* ROC *curve*. These three metrics provide good insight into a classifier's general performance and are simple to calculate. To begin, consider the multiclass case where $\mathbb{Y}$ is finite. The simplest multiclass measure is the *error rate* which is the empirical probability of a misclassification.

Definition 2.6 (Error rate). *The error rate is*

$$\text{err}(\mathcal{X}, f) := \frac{|\{(\mathbf{x}, y) \in \mathcal{X} | y \neq f(\mathbf{x})\}|}{|\mathcal{X}|}$$

The related measure accuracy is

$$\text{acc}(\mathcal{X}, f) := 1 - \text{err}(\mathcal{X}, f)$$

This measure is simple and intuitive as it is the fraction of misclassified (or correctly classified in the case of accuracy) samples, but is, unfortunately, sensitive to the *balance* of the classes; using this metric, a majority voter – a classifier which constantly predicts the class most prevalent during training despite the input – will receive an error rate of $1 - r$ if the majority class is represented with proportion r. For example, if 75% of the samples belong to the majority class, the error rate of a majority voter will be 25%. This behaviour is not always desired as no real "learning" has taken place, especially when the class proportions in $\mathcal{X}$ do not reflect the true class proportions which is typically the case in bioinformatics studies. An alternative measure that normalises for this problem is the *balanced error rate.*

Definition 2.7 (Balanced error rate)**.** *The balanced error rate is*

$$\text{balerr}(\mathcal{X}, f) := \frac{1}{|\mathbb{Y}|} \sum_{y' \in \mathbb{Y}} \text{err}(\mathcal{X}_{y'}, f)$$

where $\mathcal{X}_{y'} := \{(\mathbf{x}, y) \in \mathcal{X} | y = y'\}$*. It is the mean error rate calculated per class.*

The balanced error rate measure results in 0.5 when presented with a majority voter in the two-class case and is insensitive to class distribution. Using this metric it is easier to determine if "true learning" has taken place rather that simply learning the class distributions.

Consider now the two-class case where $\mathbb{Y} = \{1, -1\}$. Two basic metrics are the *false positive rate* (FPR) and *true positive rate* (TPR).

Definition 2.8 (True and false positive rates)**.** *The* FPR *is*

$$\text{FPR}(\mathcal{X}, f) = \frac{|\{(\mathbf{x}, y) \in \mathcal{X} | f(\mathbf{x}) = -y = 1\}|}{|\mathcal{X}|}$$

and the TPR *is*

$$\text{TPR}(\mathcal{X}, f) = \frac{|\{(\mathbf{x}, y) | f(\mathbf{x}) = y = 1\}|}{|\mathcal{X}|}.$$

The TPR *is equivalent to sensitivity and* $1 -$ FPR *is equivalent to specificity.*

Intuitively, the sensitivity measures the power to detect positive samples, and the specificity measures the power to correctly identify negative samples. Clearly

these two measures are intrinsically linked; raising sensitivity will lower specificity and vice versa. Thus, some classifiers (e.g., for prognostic tests) will require careful design to adequately balance these measures.

A useful tool that visually displays the trade-off between sensitivity and specificity is the *receiver operating characteristic* (ROC) curve. This is a plot of the FPR vs TPR while the *decision threshold* of a classifier is varied between the two extremes of majority voting. More formally, consider two-class classification with the hypothesis space $\mathcal{F} = \mathbb{R}^{\mathbb{X}}$ and class predictions $y = H \circ f$ where

$$H\colon \mathbb{R} \to \{1, -1\}$$
$$\xi \mapsto \operatorname{sign}(\xi - T) := \begin{cases} 1 & \text{if } \xi > T \\ 0 & \text{if } \xi = T \\ -1 & \text{otherwise} \end{cases}$$

for a threshold $T \in \mathbb{R}$. Here, the hypothesis f outputs a decision value or confidence score, and H maps this decision value to the classes based on the threshold T.

The threshold T may be changed to adjust the balance between the FPR and TPR. A ROC curve is produced by varying T over $\mathbb{R}$. On finite datasets, this yields a finite set of points of the ROC curve, and linear interpolation between points can be used for intermediate values. Figure 2.1 shows an example ROC curve. Here, the dashed line indicates the expectation of a trivial classifier (e.g., random guessing or a constant majority voting classifier) and the main curve has been coloured according to T.

The performance over the range of thresholds $T \in \mathbb{R}$ is highly useful information, but it is often desirable to summarise the threshold independent performance of a classifier as a single number. By calculating the *area under the* ROC *curve* (AROC or AUC, Hanley and McNeil 1982) a single threshold independent quantity is obtained which summarises the general performance.

Definition 2.9 (Area under the ROC curve (Hanley and McNeil, 1982))**.** *The area under the* ROC *curve is*

$$\text{AROC}(\mathcal{X}, f) := P(f(\mathbf{x}) > f(\mathbf{x}')|y > y') + \frac{1}{2}P(f(\mathbf{x}) = f(\mathbf{x}')|y > y')$$

It follows that the AROC estimated from a finite-sized sample is a *U-statistic*[2]

[2]A U-statistic is the sum of a function $h\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ calculated over all pairs:

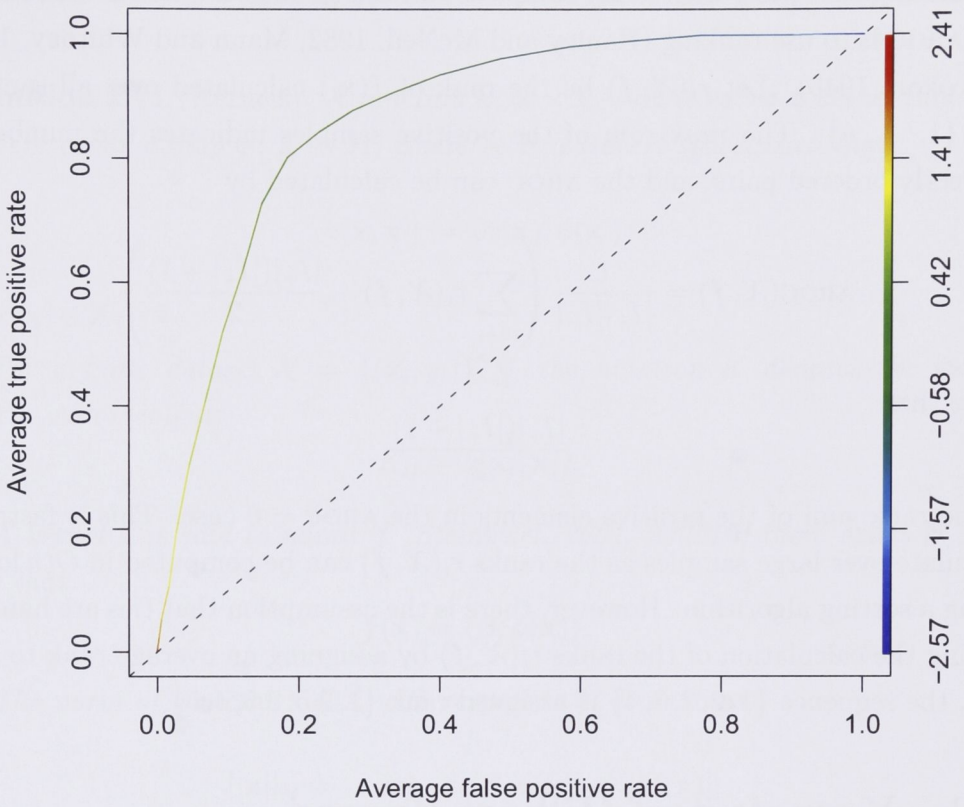


Figure 2.1: An example ROC curve. The colour of the ROC curve corresponds to the threshold indicated on the right y-axis. The diagonal dashed line is the expected performance of a trivial classifier.

that can be estimated by

$$\text{AROC}(\mathcal{X}, f) = \frac{1}{|I_+||I_-|} \sum_{i \in I_+} \sum_{j \in I_-} h(\mathbf{x}_i, \mathbf{x}_j)$$

where $I_+ = \{i | y_i = 1\}$, $I_- = \{i | y_i = -1\}$, and

$$h(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1 & \text{if } f(\mathbf{x}_i) > f(\mathbf{x}_j) \\ 0.5 & \text{if } f(\mathbf{x}_i) = f(\mathbf{x}_j) \\ 0 & \text{otherwise} \end{cases}$$

$\sum_i \sum_j h(\mathbf{x}_i, \mathbf{x}_j)$. The function h is called a *kernel function*, but should not be confused with the kernel functions associated with Hilbert spaces.

is the U-statistic kernel function.

Unfortunately, this method scales with $O(n^2)$ and so is not attractive when calculating the AROC with many samples. A more efficient method of calculating the AROC is to use ranking (Hanley and McNeil, 1982; Mann and Whitney, 1947; Wilcoxon, 1945). Let $r_i(\mathcal{X}, f)$ be the rank of $f(\mathbf{x}_i)$ calculated over all samples $i \in \{1, \ldots, n\}$. The *rank sum* of the positive samples indicates the number of correctly ordered pairs, and the AROC can be calculated by

$$\text{AROC}(\mathcal{X}, f) = \frac{1}{|I_+||I_-|}\left(\sum_{i \in I_+} r_i(\mathcal{X}, f) - \frac{|I_+|(|I_+|+1)}{2}\right).$$

Note that

$$\frac{|I_+|(|I_+|+1)}{2}$$

is the rank sum of the positive elements in the AROC $= 0$ case. This is faster to calculate over large samples as the ranks $r_i(\mathcal{X}, f)$ can be computed in $O(n \log n)$ using a sorting algorithm. However, there is the assumption that ties are handled during the calculation of the ranks $r_i(\mathcal{X}, f)$ by assigning an average rank to ties, e.g., the sequence $\{1, 2, 2, 3, 4\}$ is assigned rank $\{1, 2.5, 2.5, 4, 5\}$.

2.3 Kernels and Hilbert Spaces

Before considering how to search for a good hypothesis, the concept of Hilbert spaces and kernels is introduced. In machine learning, in particular supervised learning, the attributes of unknown samples are predicted based on the attributes of known samples. Many algorithms rely on some measure of similarity to evaluate new samples against known samples. Hilbert spaces are vector spaces equipped with an inner product, with which many geometrical concepts such as distance can be used to measure similarity. The presentation below is based on Berlinet and Thomas-Agnan (2003); Cristianini and Shawe-Taylor (2000); Schölkopf and Smola (2002); Shawe-Taylor and Cristianini (2004); Vapnik (1998, 1999).

Definition 2.10 (Hilbert space)**.** *A* Hilbert space *is a complete inner product space, that is a vector space with an inner product operator where every Cauchy sequence*[3] *converges to a point in the space, where* $||x|| := \sqrt{\langle x, x \rangle}$ *is the inner product induced norm.*

[3]Recall that a sequence $x_1, x_2, \ldots$ is called Cauchy iff $\forall \epsilon > 0$ there exists N such that $||x_n - x_m|| < \epsilon$ when $n, m \geq N$

Kernel functions allow the evaluation of inner products between two points (in $\mathbb{R}^n$) embedded in a Hilbert space without explicit calculation of the Hilbert space. This is extremely powerful as the Hilbert space itself is, in general, incalculable.

Definition 2.11 (Kernels). *A function $k\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is called a* kernel function *iff there exists a map $\phi\colon \mathbb{X} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space, such that*

$$k(\mathbf{x}, \mathbf{x}') := \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle \tag{2.2}$$

for $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$.

For a finite dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the notation K denotes the kernel matrix *with elements*

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j).$$

A kernel k is said to admit *f (Steinwart, 2001, 2005) if there exists $\boldsymbol{\beta} \in \mathcal{H}$ such that*

$$f(\mathbf{x}) = \langle \boldsymbol{\beta}, \phi(\mathbf{x}) \rangle .$$

The norm of a vector $\mathbf{x} \in \mathbb{X}$ with respect to $\mathcal{H}$ is denoted

$$||\mathbf{x}||_{\mathcal{H}} = \sup_{\boldsymbol{\beta} \in \mathcal{H}; ||\boldsymbol{\beta}|| \leq 1} \langle \boldsymbol{\beta}, \phi(\mathbf{x}) \rangle = ||\phi(\mathbf{x})||.$$

Definition 2.12 (Reproducing kernel Hilbert space). *Let $\mathcal{H}$ be a Hilbert space of real functions on $\mathbb{X}$. Then $\mathcal{H}$ is called a* reproducing kernel Hilbert space *(RKHS) iff there exists $\phi\colon \mathbb{X} \to \mathcal{H}$ such that for every $\mathbf{x} \in \mathbb{X}$*

$$f(\mathbf{x}) = \langle f, \phi(\mathbf{x}) \rangle .$$

The kernel defined by Equation 2.2 is called the reproducing kernel *for the Hilbert space $\mathcal{H}$. It follows that if k is a reproducing kernel then it has the property $\langle k(\mathbf{x}, \cdot), k(\mathbf{x}', \cdot) \rangle = k(\mathbf{x}, \mathbf{x}')$.*

Example 2.13 (The polynomial kernel). *Let $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^m$ and $d > 0$. Consider the functions with degree $d > 0$ of the form*

$$f(\mathbf{x}) := \sum_{|\boldsymbol{\alpha}| \leq d} \mathbf{u}^{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$$

and the polynomial map

$$\phi_{\mathbf{x}}(\mathbf{x}') := \sum_{|\boldsymbol{\alpha}| \leq d} \binom{d}{|\boldsymbol{\alpha}|} \mathbf{x}^{\boldsymbol{\alpha}} \mathbf{x}'^{\boldsymbol{\alpha}},$$

where $\mathbf{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} x_2^{\alpha_2} \cdots$ *and* $|\boldsymbol{\alpha}| := ||\boldsymbol{\alpha}||_1$. *We will show that* $\phi_{\mathbf{x}} \colon \mathbb{R}^m \to \mathbb{R}$ *for any* $\mathbf{x}$ *is a function in a reproducing kernel Hilbert Space.*

Define the inner product between two functions f, f' *as*

$$\langle f, f' \rangle := \sum_{|\boldsymbol{\alpha}| \leq d} \mathbf{u}^{\boldsymbol{\alpha}} \mathbf{u}'^{\boldsymbol{\alpha}} \binom{d}{|\boldsymbol{\alpha}|}^{-1}.$$

Then, the reproducing property is present:

$$\begin{aligned} \langle f, \phi_{\mathbf{x}} \rangle &= \sum_{|\boldsymbol{\alpha}| \leq d} \mathbf{u}^{\boldsymbol{\alpha}} \binom{d}{|\boldsymbol{\alpha}|} \mathbf{x}^{\boldsymbol{\alpha}} \binom{d}{|\boldsymbol{\alpha}|}^{-1} \\ &= \sum_{|\boldsymbol{\alpha}| \leq d} \mathbf{u}^{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \\ &= f(\mathbf{x}). \end{aligned}$$

Defining

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') :&= \langle \phi_{\mathbf{x}}, \phi_{\mathbf{x}'} \rangle \\ &= \sum_{|\boldsymbol{\alpha}| \leq d} \binom{d}{|\boldsymbol{\alpha}|} \mathbf{x}^{\boldsymbol{\alpha}} \binom{d}{|\boldsymbol{\alpha}|} \mathbf{x}'^{\boldsymbol{\alpha}} \binom{d}{|\boldsymbol{\alpha}|}^{-1} \\ &= \sum_{|\boldsymbol{\alpha}| \leq d} \binom{d}{|\boldsymbol{\alpha}|} \mathbf{x}^{\boldsymbol{\alpha}} \mathbf{x}'^{\boldsymbol{\alpha}} \\ &= \sum_{|\boldsymbol{\alpha}| \leq d} \binom{d}{|\boldsymbol{\alpha}|} (\mathbf{x} \bullet \mathbf{x}')^{\boldsymbol{\alpha}} \\ &= (x_1 x_2' + \cdots + x_m x_m')^d \\ &= \langle \mathbf{x}, \mathbf{x}' \rangle^d \end{aligned}$$

gives the reproducing kernel k, *where* $\mathbf{x} \bullet \mathbf{x}' = \mathbf{x}''$ *denotes the Hadamard product* $\mathbf{x}_i'' := \mathbf{x}_i \mathbf{x}_i'$. *This kernel is called the* polynomial kernel.

Example 2.14 (Gaussian Radial Basis Function Kernel (Steinwart et al., 2006)). *Let*

$$k(\mathbf{x}, \mathbf{x}') := \exp(-\sigma ||\mathbf{x} - \mathbf{x}'||_2^2)$$

for $\sigma > 0$. This kernel is known as the Radial Basis Function *(RBF) kernel. To see it is a reproducing kernel, decompose k as*

$$\exp(-\sigma \langle \mathbf{x}, \mathbf{x} \rangle) \exp(-\sigma \langle \mathbf{x}', \mathbf{x}' \rangle) \exp(2\sigma \langle \mathbf{x}, \mathbf{x}' \rangle).$$

As $\exp(2\sigma \langle \mathbf{x}, \mathbf{x}' \rangle)$ is positive definite and $\exp(-\sigma \langle \mathbf{x}, \mathbf{x} \rangle) \exp(-\sigma \langle \mathbf{x}', \mathbf{x}' \rangle)$ is positive definite, k is positive definite and hence there exists a unique Hilbert space of functions for which k is a reproducing kernel by the Moore–Aronszajn theorem (Aronszajn, 1950).

2.4 Empirical Risk Minimisation

Consider a point $(\mathbf{x}, y) \in \mathbb{X} \times \mathbb{Y}$ where $\mathbb{Y} = \mathbb{R}$ for regression problems or $\mathbb{Y} = \{1, -1\}$ for two-class problems[4]. A loss function measures the deviation of a potential predictor f at a training point $\mathbf{x}$ from observation y.

Definition 2.15 (Loss function). *A loss function is a map*

$$L \colon \mathbb{Y} \times \mathbb{R} \to [0, \infty)$$

such that

$$(y, y) \mapsto 0.$$

A good loss function for regression problems is the *least squares* loss (Hastie et al., 2001)

$$(y, f(\mathbf{x})) \mapsto (y - f(\mathbf{x}))^2. \tag{2.3}$$

This is a popular choice as the loss function is convex and easily differentiable. This generally simplifies the overall optimisation equation and an easy analytical solution can be found. An example of this is the ridge regression algorithm introduced later.

For two-class classification where $\mathbb{Y} = \{1, -1\}$, the function f indicates a degree of confidence with $\operatorname{sign} \circ f$ giving the hard classes $\{-1, 1\}$. High confidence values with the correct sign should incur no loss, while low confidence values

[4]Regression and two-class classification hypothesis functions are similar, and regression algorithms can often be used for classification with the addition of the final output transformation $y = \operatorname{sign} \circ f$.

should incur a proportional loss. A loss function that provides this is the *soft margin loss* (Boser et al., 1992; Cristianini and Shawe-Taylor, 2000; Cortes and Vapnik, 1995; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) given by

$$(y, f(\mathbf{x})) \mapsto \max(0, 1 - yf(\mathbf{x}))^p, \tag{2.4}$$

with $p > 0$. Popular instances of this loss are the *hinge loss* ($p = 1$) and *quadratic loss* ($p = 2$). Figure 2.2 illustrates these two variants.

Now that loss at a single point has been defined, the task of assessing a predictor f can be attempted. If it is assumed that the underlying probability measure is P and f is integrable, then the expected loss over this distribution can be defined.

Definition 2.16 (Expected risk). *Let $L\colon \mathbb{Y} \times \mathbb{R} \to [0, \infty)$ be a loss function, and P be the underlying probability measure generating the data. The expected risk is*

$$R[f] := E_{(\mathbf{x},y)\sim P}[L(y, f(\mathbf{x}))].$$

The expected risk cannot be directly calculated as P is unknown. Instead, the expected risk can be approximated from a finite dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ by using the *empirical density* as an approximation of P.

Definition 2.17 (Empirical risk). *Let $L\colon \mathbb{Y} \times \mathbb{R}$ be a loss function and $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a finite dataset. The empirical risk is*

$$R_{emp}[f] := \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\mathbf{x}_i)).$$

Using this framework, predictors can be induced from finite training samples by choosing a suitable loss and search for $f \in \mathcal{F}$ by minimising the empirical risk.

Example 2.18. *As an example, consider a homogeneous linear prediction function $f(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\beta} \rangle$ and choose the least squares loss (Equation 2.3). The empirical risk is then*

$$R_{emp}[f] = \sum_i (y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle)^2 = (\mathbf{y} - X\boldsymbol{\beta})^*(\mathbf{y} - X\boldsymbol{\beta}),$$

where X is a $n \times m$ matrix with each row a sample $\mathbf{x}_i$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ the vector of observations. This optimisation equation can be solved by equating the

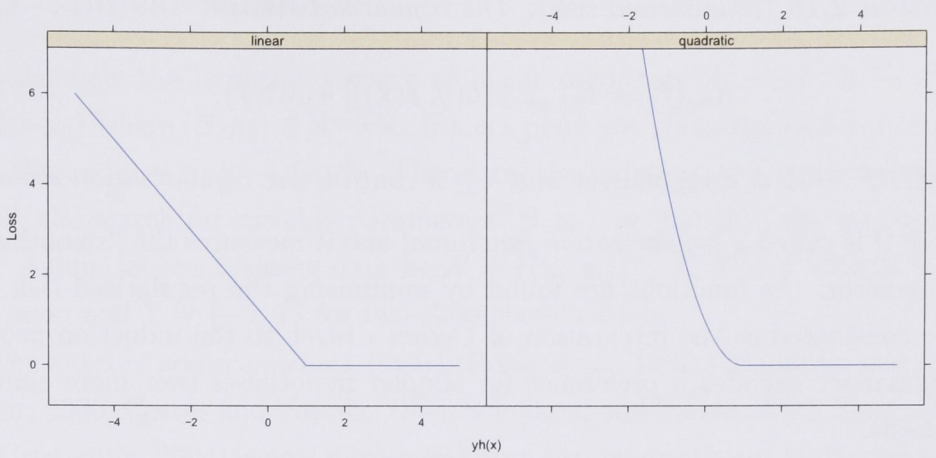


Figure 2.2: The soft-margin loss function for $p = 1$ (linear) and $p = 2$ (quadratic).

partial derivative with respect to $\boldsymbol{\beta}$ *to zero as the minimum point must also be a stationary point. Taking the partial derivative of the empirical risk with respect to* $\boldsymbol{\beta}$ *and equating to 0 gives*

$$\begin{aligned} \frac{\partial R_{emp}[f]}{\partial \boldsymbol{\beta}} &= -2X^*(\mathbf{y} - X\boldsymbol{\beta}) = 0 \\ &\Rightarrow X^*X\boldsymbol{\beta} = X^*\mathbf{y} \\ &\Rightarrow \boldsymbol{\beta} = (X^*X)^{-1}X^*\mathbf{y}, \end{aligned}$$

where the notation $(X^*X)^{-1}$ *denotes the* matrix inverse *of* X^*X*, i.e., the square matrix* B *(assuming existence) such that* $(X^*X)B = Id$*, where* Id *is the identity matrix.*

Unfortunately, the solution in the example is accompanied by several problems. First, the matrix X^*X may be singular. This can arise in many situations, including when $m > n$. This is clearly a problem in the bioinformatics field as most problems have $m \gg n$. Second, when learning by minimising the empirical risk on high-dimensional data, the problem of *overfitting* (Hastie et al., 2001) occurs; the model parameters are over-adjusted such that the model fits the training data well, but does not fit new data well. Here, the predictor has *overfit* the training data, and does not *generalise* to new data. A solution to these problems is to introduce a preference for smoother and simpler models through *regularisation.*

Definition 2.19 (Regularised risk). *The regularised risk is*

$$R_{reg}[f] := E_{(\mathbf{x},y)\sim P}[L(y, f(\mathbf{x}))] + \lambda\Omega(f),$$

where $\Omega\colon \mathcal{F} \to \mathbb{R}$ *is a* regulariser *and* $\lambda \geq 0$ *controls the regularisation strength.*

Here Ω is called a *regularisation functional* and it measures the "smoothness" of a predictor. As functions are found by minimising the regularised risk, this can be considered as the integration of *Occam's razor* to the induction process; the regulariser encodes a preference for simpler hypotheses over more complex hypothesis.

Example 2.20 (Ridge regression). *As a continuation of the previous example (Example 2.18), let us choose the* L^2 *norm as our regulariser* $\Omega(f) = ||\boldsymbol{\beta}||_2^2 = \langle \boldsymbol{\beta}, \boldsymbol{\beta} \rangle$ *(Tikhonov, 1963). As before, taking the partial derivative of* R_{reg} *(see Definition 2.19) with respect to* $\boldsymbol{\beta}$ *and equating to 0 yields*

$$\begin{aligned} R_{emp}[f] &= (\mathbf{y} - X\boldsymbol{\beta})^*(\mathbf{y} - X\boldsymbol{\beta}) + \lambda \langle \boldsymbol{\beta}, \boldsymbol{\beta} \rangle \\ \frac{\partial R_{emp}[f]}{\partial \boldsymbol{\beta}} &= -2X^*(\mathbf{y} - X\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} = 0 \\ &\Rightarrow -X^*\mathbf{y} + X^*X\boldsymbol{\beta} + \lambda\boldsymbol{\beta} = 0 \\ &\Rightarrow (X^*X + \lambda Id)\boldsymbol{\beta} = X^*\mathbf{y} \\ &\Rightarrow \boldsymbol{\beta} = (X^*X + \lambda Id)^{-1}X^*\mathbf{y}, \end{aligned}$$

where Id *is the identity matrix. Note that the inverse of* $X^*X + \lambda Id$ *exists for sufficiently small* λ *as any zero eigenvalues are shifted away from zero by* λ*, resulting in a non-zero determinant. This is the* ridge regression *solution.*

Definition 2.21 (Ridge regression). *The homogeneous ridge regression solution is given by*

$$\boldsymbol{\beta} = (X^*X + \lambda Id)^{-1}X^*\mathbf{y},$$

where I *is the identity matrix and* $\lambda > 0$ *controls the regularisation strength.*

The limit as $\lambda \to 0$ *is well defined, known as the* Moore-Penrose pseudoinverse *(Albert, 1972):*

$$X^\dagger := \lim_{\lambda \to 0} (X^*X + \lambda Id)^{-1}X^*.$$

With the established framework of risk minimisation, it is now possible to derive many of the popular classification and regression algorithms. The algorithms used in later chapters are now derived, beginning with linear methods.

2.4.1 Linear Methods

Consider now the hypothesis space of linear predictors $\mathcal{F} = \{f\colon \mathbb{X} \to \mathbb{R} | \mathbf{x} \mapsto \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0\}$ where $(\boldsymbol{\beta}, \mu_0) \in \mathbb{R}^m \times \mathbb{R}$. If $\mathbf{x}$ is a point with homogeneous coordinates[5] then this reduces to $\mathbf{x} \mapsto \langle \mathbf{x}, \boldsymbol{\beta} \rangle$. Therefore, the inducing of a linear predictor is simply the search for suitable parameters $(\boldsymbol{\beta}, \mu_0)$, or just $\boldsymbol{\beta}$ in the homogeneous case. Again, let the training data be $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{X} \times \mathbb{Y}$ with $\mathbb{Y} = \mathbb{R}$ for regression and $\mathbb{Y} = \{-1, 1\}$ for two-class classification.

The *support vector machine* (SVM) (Boser et al., 1992; Cristianini and Shawe-Taylor, 2000; Cortes and Vapnik, 1995; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) is now introduced from the perspective of *maximum margin classification*, and then shown how it can be cast in the framework of regularised risk minimisation. Consider the two-class classification case with a prediction function of the form

$$f(\mathbf{x}) = \operatorname{sign} \circ g(\mathbf{x}),$$

where

$$g(\mathbf{x}) := \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0$$

for any $\mathbf{x} \in \mathbb{X}$. This function defines a *linear hyperplane* (Rockafellar, 1997) for $g(\mathbf{x}) = 0$. Furthermore, the function $g(\mathbf{x})$ outputs the unnormalised distance of the sample from this hyperplane. This can be interpreted as an uncalibrated measure of confidence; the larger $|g(\mathbf{x})|$ then the further $\mathbf{x}$ lies from the decision boundary and thus the more confidence of a correct prediction.

The previously introduced method of ridge regression can be applied on two-class data to find suitable parameters $(\boldsymbol{\beta}, \mu_0)$. This application of ridge regression on classification data is also known as *regularisation networks*. Though this method can perform well, it may not be optimal as the least squares loss term will penalise highly confident predictions. What is desired is a hyperplane that cleanly separates the two classes, i.e., such that $yg(\mathbf{x}) \geq 1 \ \forall (\mathbf{x}, y) \in \mathcal{X}$, and a loss function that does not penalise incorrect classifications. A hyperplane that separates the classes perfectly is called a *separating hyperplane*.

Let the points in the data that satisfy $yg(\mathbf{x}) = 1$ be called *support vectors* and consider the distance between two support vectors of opposite classes, that is a $\mathbf{x}$ and $\mathbf{x}'$ such that

$$g(\mathbf{x}) = 1 \ \& \ g(\mathbf{x}') = -1.$$

[5]The homogeneous coordinates of $\mathbf{x} = (x_1, \ldots, x_m)$ here indicates the $(m+1)$-tuple $\mathbf{x}' = (x_1, \ldots, x_m, 1)$. This allows affine transformations to be represented by matrix products.

This distance is called the *margin* and is given by

$$\frac{\langle \mathbf{x}, \boldsymbol{\beta} \rangle}{||\boldsymbol{\beta}||_2^2} - \frac{\langle \mathbf{x}', \boldsymbol{\beta} \rangle}{||\boldsymbol{\beta}||_2^2} = \frac{2}{||\boldsymbol{\beta}||_2^2}.$$

An illustration can be seen in Figure 2.3.

The *hard-margin support vector machine* is a classifier that seeks a separating hyperplane with maximal margin.

Proposition 2.22 (Hard-margin support vector machine (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004))**.** *Given a finite dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the hard-margin support vector machine is given by the solution to the optimisation problem*

$$\min \frac{1}{2}||\boldsymbol{\beta}||_2^2 \tag{2.5}$$
$$\text{such that } \forall i \; y_i(\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \mu_0) \geq 1.$$

This is known as the primal problem.

Example 2.23 (The dual problem)**.** *Finding the solution for the hard-margin* SVM *is generally done for the* dual problem *not the* primal problem*. This example will derive the dual problem using Lagrange optimisation (Nocedal and Wright, 2006). Introducing the multipliers α_i for the constraints in* (2.5) *gives the Lagrangian function*

$$\Phi(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\alpha}) := \frac{1}{2}||\boldsymbol{\beta}||_2^2 + \sum_i \alpha_i (1 - y_i(\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \mu_0)). \tag{2.6}$$

The saddle point is stationary, thus equating the partial derivatives of (2.6) *with respect to the parameters to 0 gives the conditions*

$$\frac{\partial \Phi}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_i \alpha_i y_i \mathbf{x}_i = 0$$
$$\Rightarrow \boldsymbol{\beta} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{2.7}$$
$$\frac{\partial \Phi}{\partial \mu_0} = \sum_i \alpha_i y_i = 0. \tag{2.8}$$

Note that (2.7) *shows that $\boldsymbol{\beta}$ is a linear combination of the samples, and $\boldsymbol{\alpha}$ and $\mathbf{y}$ are orthogonal vectors by* (2.8)*. Substituting* (2.7) *and* (2.8) *into* (2.6) *and*

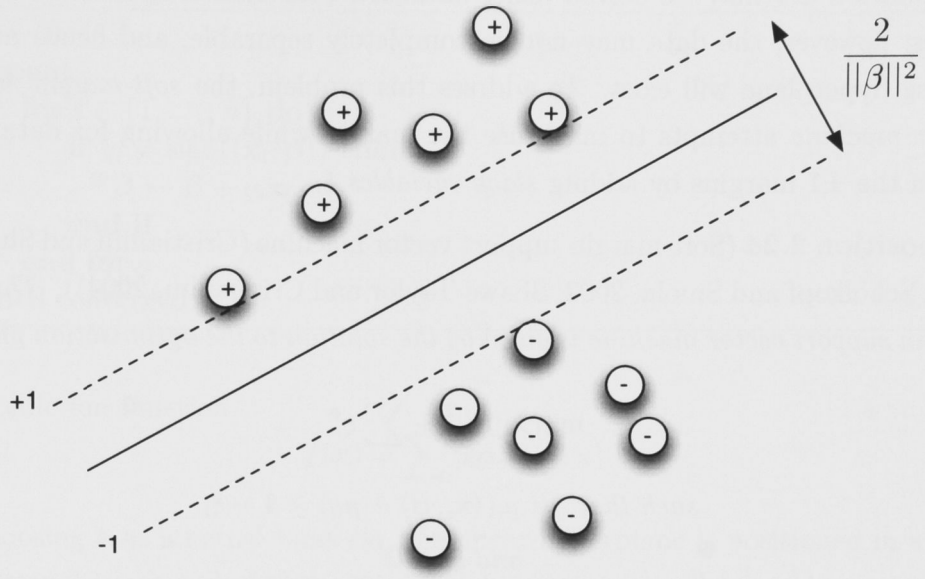


Figure 2.3: Illustration of some 2-class data (the + and − marked circles) and a separating hyperplane (the solid line). The margin is indicated by the dashed lines.

simplifying gives

$$\begin{aligned}\Phi(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\alpha}) &= \frac{1}{2}||\boldsymbol{\beta}||_2^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle - \mu_0 \sum_i \alpha_i y_i \\ &= \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + \sum_i \alpha_i - \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &=: \Upsilon(\boldsymbol{\alpha}).\end{aligned}$$

The dual optimisation problem *is thus*

$$\max_{\alpha_i \geq 0} \Upsilon(\boldsymbol{\alpha}) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \qquad (2.9)$$
$$\text{such that} \quad \sum_i \alpha_i y_i = 0.$$

Note that $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ *in* (2.9) *may be replaced by a kernel function* $k(\mathbf{x}_i, \mathbf{x}_j)$. *Doing so positions the hyperplane in a high (potentially infinite) dimensional Hilbert space, allowing the hard-margin* SVM *to learn non-linear functions.*

Equation 2.9 may be solved using standard constrained optimisation techniques; however, the data may not be completely separable, and hence no separating hyperplane will exist. To address this problem, the *soft-margin support vector machine* attempts to maximise the margin while allowing for data lying within the ± 1 margins by adding *slack variables* ξ_i.

Proposition 2.24 (Soft-margin support vector machine (Cristianini and Shawe-Taylor, 2000; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004)). *The soft-margin support vector machine is given by the solution to the optimisation problem*

$$\begin{gathered} \min \lambda ||\boldsymbol{\beta}||_2^2 + \sum_i \xi_i^p \\ \textit{such that } \forall i \; y_i(\langle \mathbf{x}_i, \boldsymbol{\beta} \rangle + \mu_0) \geq 1 - \xi_i \\ \textit{and } \xi_i \geq 0 \end{gathered}$$

for $p \geq 1$.

The final classifier to be introduced here is *Rosenblatt's perceptron*. This is again a separating hyperplane classifier (not maximum margin) and has a simple iterative training procedure. The convergence of the algorithm for separable data is ensured by Novikoff's theorem (Cristianini and Shawe-Taylor, 2000). Like the hard-margin SVM, when the data is not separable there is no solution and the algorithm will not converge. In such a case, some other criteria must be used for terminating the infinite loop (such as a maximum number of iterations).

The perceptron may be extended to fit non-linear functions by "kernelising" the linear perceptron algorithm. This gives rise to the kernel-perceptron (Cristianini and Shawe-Taylor, 2000; Freund and Schapire, 1999). Let $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ be a finite dataset. It follows directly from the linear perceptron algorithm that the solution lies in the span of the training samples, hence there exists an n-tuple $\boldsymbol{\alpha} \in \mathbb{N}^n$ such that the solution is

$$\boldsymbol{\beta} = \sum_i \alpha_i y_i \mathbf{x}_i,$$

where $(\mathbf{x}_i, y_i) \in \mathcal{X}$. The prediction function for new sample $\mathbf{x}$ is then

$$f(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\beta} \rangle = \left\langle \mathbf{x}, \sum_i \alpha_i y_i \mathbf{x}_i \right\rangle = \sum_i y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle .$$

The inner product $\langle \mathbf{x}_i, \mathbf{x} \rangle$ can be replaced by a kernel function $k(\mathbf{x}_i, \mathbf{x})$, producing

Algorithm 2.1 Rosenblatt's perceptron

```
β ← 0
repeat
    for i ∈ {1, ..., n} do
        if y_i ≠ sign(⟨x_i, β⟩) then
            β ← β + y_i x_i
        end if
    end for
until converged
```

the prediction function

$$f(\mathbf{x}) = \sum_i y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}).$$

By choosing k as a kernel function, the linear hyperplane is positioned in a high dimensional space rather than the direct feature space, allowing the perceptron to learn non-linear functions.

2.4.2 Non-linear Methods

In the previous section, it was noted that the SVM solutions lie in the span of the support vectors. This is especially clear in the dual optimisation problem. It was remarked that the dot product arising there may be replaced with a non-linear kernel function to provide the ability to learn non-linear problems. Suppose $\Phi\colon \mathbb{X} \to \mathcal{H}$ is a non-linear mapping to a Hilbert space with kernel function $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$ and kernel matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. When the kernel matrix K is symmetric and positive semi-definite it can be decomposed into a lower-triangular matrix L such that $K = LL^*$ using the Cholesky decomposition. Any linear method may then be extended into non-linearity by using this decomposition L as a substitution for data matrix X due to the mathematical equivalence of the linear dot product between vectors of L and the kernel function.

Although useful for some problems, in bioinformatics these *kernel methods* are of limited use due to a couple of problems. First, the increase in fitting ability can lead to problems of overfitting. This is especially worrying in bioinformatics applications as the sample size is very small and simple linear classifiers can already show signs of overfitting. Second, the mapping to high-dimensional space tends to "mask" features that are useful for prediction. Interpretable models may not be required for some applications of machine learning (e.g., handwriting recognition), but it is in the bioinformatics domain as frequently one wishes to

Algorithm 2.2 Kernel perceptron

```
1: α ← 0
2: repeat
3:     for i ∈ {1, ..., n} do
4:         if y_i ≠ sign(Σ_j y_j α_j k(x_j, x_i)) then
5:             α_i ← α_i + 1
6:         end if
7:     end for
8: until converged
```

gain insight into the underlying biological process. These issues restrict most kernel methods to pure classification applications in bioinformatics.

2.5 Feature Selection

Feature selection is a technique for reducing the size of models when mining data for information. Frequently, the features needed for good performance of the predictor are desired. Locating these features allows further biological experiments to be designed, and may lead to identification of important genes or genetic regions. Other benefits of feature selection may be increased prediction performance due to the elimination of noisy features.

Feature selectors may be broadly classified into three classes: filters, wrappers, and embedded methods (Guyon, 2003; Guyon et al., 2006). Filters operate on the data independently of the predictor in use. Wrappers treat the predictor as a black box, evaluating the performance of different feature sets by repeated retraining and prediction. Finally, embedded methods include feature selection as part of the training procedure.

2.5.1 Filters

Filters operate on the data without any consideration for the type of predictor. This typically is not ideal as the feature subsets will not be "tailored" for the classifier, and better performance may be obtained with a different subset. Another problem among filters is the selected subsets tend to contain highly correlated features. This may lead to sub-optimal performance as orthogonal features may provide more information. Of course, in the case of noisy data – which microarray data certainly is – higher numbers of correlated features may lead to better predictor stability and hence classification. Because of this, achieving a good

balance between correlated and orthogonal features is important, though most filters ignore the latter.

Consider first independent ranking feature selectors. These filters assign a score for each feature independently, with features then being selected from best to worst. More formally, suppose each feature is assigned a score υ_i by the scoring function. An ordering $\Upsilon := (|\upsilon|, \geq)$ is determined, where $\upsilon = \{\upsilon_i\}$, and the first $m_{\text{feats}} \leq m$ elements are selected.

Recall that for a finite dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the matrix representation is X where the i^{th} row is a sample $\mathbf{x}_i$ and the j^{th} column is a *feature vector* $\mathbf{x}^{(j)}$. For regression problems an obvious choice is Pearson's correlation:

$$\upsilon_i = \text{cor}(\mathbf{x}_c^{(i)}, \mathbf{y}_c) := \frac{\left\langle \mathbf{x}_c^{(i)}, \mathbf{y}_c \right\rangle}{||\mathbf{x}_c^{(i)}||_2 ||\mathbf{y}_c||_2}$$

where $\mathbf{x}_c^{(i)} = \mathbf{x}^{(i)} - \frac{1}{n}\sum_j \mathbf{x}^{(j)}$ and $\mathbf{y}_c = \mathbf{y} - \frac{1}{n}\sum_i y_i$ are *centred* versions of $\mathbf{x}^{(i)}$ and $\mathbf{y}$.

For classification, many filters fall under the framework

$$\upsilon_j = \frac{\frac{1}{|I_+|}\sum_{i \in I_+} x_{ij} - \frac{1}{|I_-|}\sum_{i \in I_-} x_{ij}}{\sigma_j},$$

where $I_+ = \{i | y_i = 1\}$, $I_- = \{i | y_i = -1\}$, and $\sigma_j \in \mathbb{R}$ is a feature-dependent scaling factor, typically chosen to be an estimate of the standard deviation of feature j. This score measures the scaled distance between the two class centroids along a given dimension. A feature is thus considered more useful if the centroids are better separated, a characteristic that is highly intuitive for classification given the principle of maximum margin classifiers covered earlier.

Clearly the differences arise from the choice of σ_j, with the simplest choice being $\sigma_j = \sigma_{j'}$ for all $j, j' \in \{1, \ldots, m\}$, completely ignoring the variance differences among features. In the special case where the dataset is log transformed, this is also known as the *log fold change*. This measure is not so desirable if the variance varies across the features considerably, but it can also be better as variance estimates in high-dimensional spaces can be inaccurate.

The next simplest variance measure is $\sigma_j^2 = (\sigma_j^+)^2 + (\sigma_j^-)^2$, where

$$(\sigma_j^+)^2 := \frac{1}{|I_+| - 1} \sum_{i \in I_+} (x_{ij} - \frac{1}{|I_+|} \sum_{k \in I_+} x_{kj})^2 \text{ and}$$
$$(\sigma_j^-)^2 := \frac{1}{|I_-| - 1} \sum_{i \in I_-} (x_{ij} - \frac{1}{|I_-|} \sum_{k \in I_-} x_{kj})^2.$$

This is known as the *signal to noise ratio* (SNR) and was popularised by Golub et al. (1999).

From classical statistics, Student's *t-test* can be used to test for differential expression. This corresponds to the choice

$$\sigma_j = \sqrt{\frac{(\sigma_j^+)^2}{|I_+|} + \frac{(\sigma_j^-)^2}{|I_-|}}.$$

As this measure is supported by classical statistical theory, a null-hypothesis test can be carried out using the t-distribution to determine p-values. However, in bioinformatics it is commonplace to rank the features using the t-test and simply select the highest ranked m_{feat} instead of performing null-hypothesis testing.

Recently, *moderated t-statistics* (Smyth, 2004; Tusher et al., 2001) have been suggested as replacements for the t-test when using high-dimensional microarray data. These moderated t-statistics are designed to overcome a major problem of the ordinary t-test: a large t-test value may result if the variances are small even if the class separation is small. These features are not deserving of their score as good separation is required for good class discrimination. A recent solution is to add a small constant to the variance estimate, thus preventing the denominator from reaching critically small values:

$$\sigma_j := \delta + \sqrt{\frac{(\sigma_j^+)^2}{|I_+|} + \frac{(\sigma_j^-)^2}{|I_-|}}$$

where $\delta > 0$. This can be viewed as a form of *regularised* t-test, where δ is the regularisation constant. The two extremes when $\delta = 0$ and $\delta \to \infty$ represent the ordinary t-test and log fold change situations.

Now there is the additional problem of how to choose δ sensibly. Tusher et al. (2001) chose δ by minimising the t-statistic variance across different subsets of data:

$$\hat{\delta} = \arg\min_{\delta} \text{var}(\{v^{\mathcal{X}^1}, v^{\mathcal{X}^2}, \ldots\})$$

where $\mathcal{X}^1, \mathcal{X}^2, \ldots \subset \mathcal{X}$ and $v^{\mathcal{X}^i}$ is the statistic calculated using the subset $\mathcal{X}^i$. This method is entitled *significance of microarrays* (SAM). A more structured approach was proposed by Smyth (2004) under the name *linear models for microarray analysis* (LIMMA) whereby certain assumptions are made about the distribution of microarray data, allowing an estimate of δ to be derived analytically. However, in SAM's case δ conditions the standard deviation whereas the δ in LIMMA's case conditions the variance:

$$\sigma_j = \sqrt{\delta + \frac{(\sigma_j^+)^2}{|I_+|} + \frac{(\sigma_j^-)^2}{|I_-|}}.$$

Thus there is a functional difference between LIMMA and SAM.

From information theory, a popular feature filter is *mutual information*. This is a special case of *Kullback-Leibler divergence* (KL-divergence).

Definition 2.25 (Kullback-Leibler divergence). *The* KL*-divergence between two probability distributions P and Q continuous on some measure μ with densities $dP = pd\mu$ and $dQ = qd\mu$ is*

$$\int_{\mathcal{X}} p \log \frac{p}{q} d\mu$$

For discrete samples this reduces to

$$\sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$$

The KL-divergence measures the difference between two probability distributions, though it is not a metric as the symmetry requirement does not hold. Using the KL-divergence, the difference between the joint and product probability distributions of a particular feature can be measured. This is the *mutual information*.

Definition 2.26 (Mutual information). *The mutual information of a feature $\mathbf{x}^{(j)}$ is*

$$\sum_j \sum_y P(\mathbf{x}^{(j)}, y) \log \frac{P(\mathbf{x}^{(j)}, y)}{P(\mathbf{x}^{(j)})P(y)}$$

If there is no dependence between $\mathbf{x}^{(j)}$ and y, then the joint probability distribution $P(\mathbf{x}^{(j)}, y)$ is equivalent to $P(\mathbf{x}^{(j)})P(y)$ and the mutual information will

be 0. Using the mutual information, each feature may be given a score

$$v_j = \sum_{(\mathbf{x},y)\in\mathcal{X}} P(\mathbf{x}^{(j)}, y) \log \frac{P(\mathbf{x}^{(j)}, y)}{P(\mathbf{x}^{(j)})P(y)}$$

and features selected as in the previous section.

The difficulty of applying this feature selector lies in the estimation of $P(\mathbf{x}, y)$, $P(\mathbf{x})$, and $P(y)$. In the rare case of discrete values (i.e., $\mathcal{X} \subset \mathbb{Z}^m \times \mathbb{Z}$), the distributions may be estimated from frequency counts. In the case of two-class classification, the joint distribution $P(\mathbf{x}, y)$ can be written $\mathcal{X} \subset \mathbb{X} \times \mathbb{Z}$,

$$P(\mathbf{x}, y) = P(\mathbf{x}|y)P(y)$$

with $P(y)$ being estimated from frequency counts, and $P(\mathbf{x}|y)$ from a histogram or using Parzen windows density estimation (Hastie et al., 2001).

2.5.2 Wrappers

Unlike filters, wrappers (Guyon, 2003) consider the predictor used and tailor the selected features towards the predictor in use. They directly use the classifier to optimise the feature set by maximising generalisation error, while treating the predictor purely as a "black box." Typically, resampling procedures (see Section 2.6) are used to estimate the generalisation error of a predictor with a given metric for a variety of different feature subsets.

Exhaustive testing of all feature subsets is not computationally feasible with more than a few features and restrictions to the search space must therefore be made. A common approach to this problem is to use *nested subset selection* (Guyon, 2003) where nested subsets are formed by greedily adding or removing features. There are two main greedy approaches, *forward selection* and *recursive feature elimination* (RFE). For both methods, the restriction of the search path to nested subsets drastically reduces the search space to a search of complexity $O(m^2)$ where m is the number of features.

RFE begins with the whole feature set. Each feature is then discarded, creating m sets of $m-1$ features, and the generalisation performance of the chosen predictor is estimated for each feature set using resubstitution. The feature set with the best performance is then selected, and the RFE procedure repeated on this set. This produces a nested sequence of feature sets $S_1 = \{1, \ldots, m\} \supset S_2 \supset S_3 \cdots \supset S_{m+1} = \emptyset$ as the recursion progresses.

Forward selection commences with an empty set of features and gradually adds features. For the first addition, m feature sets containing a single feature are evaluated via resubstitution, and the feature set with the best generalisation performance chosen. The procedure is repeated adding another feature to the set and evaluating the generalisation performance. This results in another nested sequence of sets $S_1 = \emptyset \subset S_2 \subset S_3 \cdots \subset S_{m+1} = \{1, \ldots, m\}$.

2.5.3 Embedded methods

Embedded methods (Guyon, 2003) are between the two extreme behaviours of filters and wrappers; they do not ignore the classifier, nor do they treat it as a black box. Embedded methods use knowledge about the particular predictor chosen to help in the selection of features useful for that method. The simplest embedded methods use the RFE and forward selection procedures of the previous section, but use knowledge of the predictor to accelerate computational performance. Instead of estimating the generalisation performance using a resampling procedure for each potential addition or deletion it is estimated directly from the model, thus models only need to be trained for a change in model size, which reduces the complexity to $O(m)$.

For RFE with non-homogeneous linear models, the prediction function is $f(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0$. If each dimension is scaled such that they are directly comparable – i.e., if the data is centred and scaled to a standard deviation of 1 – then $|\beta_j|$ is an estimate of the importance of feature j. Thus at each iteration, the feature $\hat{\jmath} = \arg\min_j |\beta_j|$ is discarded and the model retrained. The specific combination of recursive feature elimination applied to support vector machines is called RFE–SVM and was first proposed by Guyon et al. (2002).

As for forward selection with linear models, the residual error can be calculated $r := y - f(\mathbf{x})$ and the best feature to add is simply the feature most correlated with r, $\hat{\jmath} = \arg\max_j |\operatorname{cor}(\mathbf{x}^{(j)}, r)|$. After adding the feature, the model can be retrained and a new residual vector calculated.

Recall the regularised empirical risk functional defined earlier:

$$R_{\text{emp}}[f] = \frac{1}{n} \sum_i L(y_i, f(\mathbf{x}_i)) + \lambda \Omega(f)$$

Consider the choice of regulariser $\Omega(f)$. Previously, $\Omega(f) = ||\boldsymbol{\beta}||_2^2$ was chosen as the regulariser as it provided some advantages. First, it is a convex regulariser

thus creating a convex optimisation problem when paired with a convex loss function. Second, it is continuously differentiable. Finally, it provides a unique solution to the problem in the underdetermined case ($m > n$). What it does not do is restrict the size of the model, or the number of non-zero elements of $\boldsymbol{\beta}$. This can be achieved by choosing the regularisation functional as $\Omega(f) = ||\boldsymbol{\beta}||_0$. Clearly, this results in the number of features being restricted with the number of features being indirectly specified by λ. Unfortunately, it is not continuously differentiable, thus the solution is difficult to find (Candès and Tao, 2005).

Therefore, an approximation that is easily solvable and encodes a preference towards sparsity is needed. First, note why the L^2 regulariser $\Omega(f) = ||\boldsymbol{\beta}||_2^2$ does not lead to sparsity. Consider two perfectly correlated features i and j and let the total mass be T. The penalty if the entire mass is assigned to one feature – either i or j – is T^2. However, the mass can be split evenly over both i and j as they are perfectly correlated, in which case the penalty is $2(\frac{T}{2})^2 = \frac{T^2}{2} < T^2$. Thus, the regulariser favours spreading the mass out among correlated features rather than removing correlated features from the model. Although this may be a good prospect in terms of stability, it is not desired behaviour when seeking sparse models. Consider the same two features under the L^1 penalty $\Omega(f) = ||\boldsymbol{\beta}||_1$. Here, the penalty is T regardless of the distribution between the two features, so this regulariser does not encourage spreading the mass between correlated features. Eliminating this preference for spreading the mass allows some elements to shrink to 0 (Candès and Tao, 2005; Donoho et al., 2005; Wainwright, 2006), and hence provides the sparsity required. Though the L^1-norm is not continuously differentiable, the optimisation problem is still tractable and can be solved using either quadratic or linear programming, depending on the chosen loss function L. This regulariser is known in other domains as the *lasso* (Tibshirani, 1996) and *basis pursuit* (Chen et al., 1998).

2.6 Estimating the Generalisation Error

Several metrics have been presented allowing the measure of prediction performance of a given model, but so far no discussion has been given on how to measure the performance in an unbiased way. One cannot use the same data used for training the predictor to measure the performance, as this estimate will be biased as any sufficiently complex model can fit the data perfectly. What is of primary interest is not this training performance, but rather the *generalisation performance*

of a model. This is the expected performance on new data unseen during the fitting of the model, and so samples need to be withheld from the training of the classifier for estimating the generalisation error. These estimation methods are commonly known as *resampling methods.*

The three main resampling techniques are bootstrapping (Efron, 1983, 1986; Efron and Tibshirani, 1994, 1997; Hastie et al., 2001), cross-validation (Hastie et al., 2001), and repeated hold-out. Of the three, repeated hold-out is the simplest. One simply randomly selects $n_{\text{train}} < n$ samples for training, trains the predictor on these samples, and measures the metric on the remaining. This is repeated several times to observe the distribution of the metric. The remaining two methods will be expounded in the following sections as they form the basis of other chapters.

2.6.1 The Bootstrap

The bootstrap is a classical statistical procedure that resembles repeated-hold out, differing mainly in how the training set is created. The three main methods of bootstrap estimation are the ϵ-0, .632, and .632+ estimators. To commence, the ϵ-0 is introduced as it is the simplest bootstrap estimator.

Like repeated hold-out, the ϵ-0 bootstrap creates two sets $\mathcal{X}_{\text{train}} \subset \mathcal{X}$ and $\mathcal{X}_{\text{test}} = \mathcal{X} \setminus \mathcal{X}_{\text{train}}$ with $\mathcal{X}_{\text{test}}$ being reserved for estimating performance and $\mathcal{X}_{\text{train}}$ for inducing a predictor. The difference lies in how the set $\mathcal{X}_{\text{train}}$ is created. In repeated hold-out, $n_{\text{train}} < n$ samples are randomly selected without replacement from $\mathcal{X}$ to form $\mathcal{X}_{\text{train}}$. In the ϵ-0 bootstrap, $n_{\text{train}} = n$ samples are selected randomly *with replacement* from $\mathcal{X}$ to form $\mathcal{X}_{\text{train}}$. By doing this, the dataset $\mathcal{X}$ is treated as the whole population which is then sampled to produce a training set of the same size. As the sampling is with replacement, there will be duplicated samples in the training set, and samples not in the training set that form the test set $\mathcal{X}_{\text{test}} = \mathcal{X} \setminus \mathcal{X}_{\text{train}}$. This is easily seen as the probability of a sample belonging to the training set is

$$\begin{aligned} P(i \in \mathcal{X}_{\text{train}}) &= 1 - (1 - \frac{1}{n})^n \\ &\approx 1 - e^{-1} \\ &\approx 0.632 \end{aligned}$$

This bootstrapping procedure is repeated several times – 100 iterations is typical

– forming a distribution for the test metric S.

Let $\mathcal{B} = \{(\mathcal{X}^i_{\text{train}}, \mathcal{X}^i_{\text{test}})\}_{i=1}^{n_\mathcal{B}}$ be a set of training and testing sets such that $\mathcal{X}^i_{\text{train}} \subset \mathcal{X}$ is formed by random sampling with replacement of $\mathcal{X}$ and $\mathcal{X}^i_{\text{test}} = \mathcal{X} \setminus \mathcal{X}^i_{\text{train}}$. Furthermore, let $f_i \colon \mathcal{X} \to \mathbb{Y}$ be the predictor induced on the set $\mathcal{X}^i_{\text{train}}$.

Definition 2.27 (ϵ-0 bootstrap estimator). *The ϵ-0 bootstrap estimate of the* GOF *measure L is*

$$\widehat{Err}_{\epsilon 0} := \frac{1}{n_\mathcal{B}} \sum_{i=1}^{n_\mathcal{B}} L(\mathcal{X}^i_{test}, f_i)$$

A problem with the ϵ-0 bootstrap estimator is that the estimate $\widehat{\text{Err}}_{\epsilon 0}$ is pessimistically biased due to the *learning curve effect* – a predictor is expected to perform worse on smaller training datasets due to less information. The .632 bootstrap is a variant of the ϵ-0 estimator that attempts to address this problem. It is a heuristic based on the observation that the average proportion of samples in the training set is 0.632. Based on this fraction, the whole dataset estimate of S can be mixed in to produce a less biased estimate.

Definition 2.28 (.632 bootstrap estimator). *The .632 bootstrap estimate of the* GOF *measure L is*

$$\widehat{Err}_{.632} = 0.632 \times \widehat{Err}_{\epsilon 0} + 0.368 \times \overline{Err},$$

where

$$\overline{Err} = \frac{1}{n_\mathcal{B}} \sum_{i=1}^{n_\mathcal{B}} L(\mathcal{X}^i_{train}, f_i).$$

Note that $\overline{\text{Err}} \leq \widehat{\text{Err}}_{.632} \leq \widehat{\text{Err}}_{\epsilon 0}$ with high probability due to the training bias in $\overline{\text{Err}}$.

A problem with this estimator is that the estimate is too optimistic for overfit predictors. Let us assume one has a two-class classifier that fits the training data correctly ($\overline{\text{Err}} = 0$), but incorrectly classifies all new data samples ($\widehat{\text{Err}}_{\epsilon 0} = 0.5$). The correct estimated generalisation error rate should be 0.5 as the classifier has not learnt any structure within the data and has just overfit the training samples. However, the .632 estimate as defined above is $\widehat{\text{Err}}_{.632} = 0.632 \times 0.5 = 0.316$, substantially lower than the true generalisation performance of 0.5.

The .632+ bootstrap estimator addresses this bias by adjusting the mixing based on the estimated amount of overfitting.

Definition 2.29 (.632+ bootstrap estimator). *The .632+ bootstrap estimate of the* GOF *measure L is*

$$\widehat{Err}_{.632+} = \delta \times \widehat{Err}_{\epsilon 0} + (1 - \delta) \times \overline{Err},$$

where

$$\delta = \frac{.632}{1 - .368R}$$

$$\text{and } R = \frac{\widehat{Err}_{\epsilon 0} - \overline{Err}}{\gamma - \overline{Err}}.$$

The parameter γ is defined as $\gamma := L(\Lambda, h)$ where h is the predictor induced on the set $\mathcal{X}$, and Λ is the set containing all possible permutations of $\mathbf{x}_i$ and y_i in $\mathcal{X}$. It is called the no-information error rate *and estimates the expected performance of a hypothesis class on data with no information (independent labels and samples).*

In this bootstrap method, $R \in [0, 1]$ is an estimate of the *overfitting rate* and $\delta \in [.632, 1]$ adjusts the mixing between the whole dataset error and the independent testing error. The overfitting rate is calculated by comparison with the no-information error rate γ, calculated by permuting the labels. When the classifier has overfit the training data $R = 1$ and $\delta = 1$, and conversely when the predictor has not overfit the training data $R = 0$ and $\delta = .632$. Thus this method allows adaptation between the .632 and ϵ-0 estimates based on the estimated amount of overfitting.

2.6.2 Cross-Validation

Cross-validation is another resampling method that can be used to estimate the generalisation error. k-fold cross-validation involves dividing data into k non-intersecting sets. Each set is then withheld for testing the classifier trained on the remaining data. As independent data is used for testing the classifier, a more accurate estimate of the generalisation performance is obtained like the bootstrap, but as k is usually small (< 10) the runtime is significantly less.

Definition 2.30 (k-fold cross-validation). *Let $\mathcal{K} = \{\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_k\}$ be the divi-*

sion into k-sets such that

$$\mathcal{K}_i \subset \mathcal{X}$$
$$\mathcal{K}_i \cap \mathcal{K}_j = \emptyset$$
$$|\mathcal{K}_i| \approx |\mathcal{K}_j|$$
$$\bigcup_i \mathcal{K}_i = \mathcal{X}$$

for all i, j. *The k-fold cross-validation estimate of* GOF *measure* L *is*

$$Err_{CV} = \frac{1}{k} \sum_{i=1}^{k} L(K_i, f^{\mathcal{X} \setminus K_i}),$$

where $f^{\mathcal{X} \setminus K_i} \colon \mathbb{X} \to \mathbb{Y}$ *is a classifier trained on the set* $\mathcal{X} \setminus K_i$.

There are various ways of assigning the data, forming set $\mathcal{K}$. The simplest approach is to randomly assign each sample – irrespective of its label – to a fold uniformly. For classification, this type of assignment is not good due to the introduction of *stratification bias* (Parker et al., 2007). In this case, a stratified fold assignment which distributes the samples per class uniformly across each k folds is better as it reduces the stratification bias to low levels. A variance estimate for Err_{CV} can be obtained by repeating the k-fold cross-validation procedure.

The special case when $k = n$ is known as the *leave one out* (LOO) estimate and is quite popular when the dataset is small and has the attractive advantage of being deterministic, however it can suffer significantly from stratification bias (Parker et al., 2007). A problem with LOO is that it cannot estimate the AROC in an unbiased way as the empirical AROC estimate requires at least one sample from each class. Consequentially, to calculate AROC using LOO one needs to "collect" each prediction for each sample and calculate AROC over the entire dataset. More concisely, let $f_i \colon \mathbb{X} \to \mathbb{Y}$ be the predictor obtained after training on all samples excluding sample i. The LOO-AROC estimate[6] is then

$$\text{LOO-AROC}(\mathcal{X}, \{f_i\}) := \frac{1}{|I_+||I_-|} \sum_{i \in I_+} \sum_{j \in I_-} \begin{cases} 1 & \text{if } f_i(\mathbf{x}_i) > f_j(\mathbf{x}_j) \\ 0.5 & \text{if } f_i(\mathbf{x}_i) = f_j(\mathbf{x}_j) \\ 0 & \text{otherwise} \end{cases}$$

[6]Note this is a heuristic and not an estimator in the strict sense of convergence. It is, however, a frequently used technique (Witten and Frank, 2005).

where $I_+ = \{i|y_i = 1\}$ and $I_- = \{i|y_i = -1\}$. To illustrate the source of the bias, suppose the predictor is a majority voter and the dataset has an equal representation of both classes. Then, as the class proportions between training and testing subset pairs are *negatively correlated*, then the AROC estimate will be below 0.5 and may be mistaken for *anti-learning* (see Chapter 6).

An estimation technique related to LOO specifically targeted towards estimating the AROC is the *leave two out* (LTO) estimator. Instead of withholding only one sample, one sample from each class is withheld. Let $f_{ij}: \mathbb{X} \to \mathbb{R}$ be the predictor obtaining from training on $\mathcal{X}$ excluding samples i and j. The LTO-AROC estimate is then

$$\text{LTO-AROC}(\mathcal{X}, \{f_{ij}\}_{i\in I_+, j\in I_-}) := \frac{1}{|I_+||I_-|} \sum_{i\in I_+} \sum_{j\in I_-} \begin{cases} 1 & \text{if } f_{ij}(\mathbf{x}_i) > f_{ij}(\mathbf{x}_j) \\ 0.5 & \text{if } f_{ij}(\mathbf{x}_i) = f_{ij}(\mathbf{x}_j) \\ 0 & \text{otherwise} \end{cases}.$$

This estimator is not affected by the aforementioned stratification bias as there is no correlation between class proportions in the training and testing subsets.

2.6.3 Comparing Cross-Validation and the Bootstrap

When comparing cross-validation methods and bootstrap methods, one finds the former to have less bias but the latter to have less variance (Braga-Neto and Dougherty, 2004; Efron, 1983; Efron and Tibshirani, 1997). An illustration of this can be seen in Figure 2.4. For this experiment, the AROC was calculated using each method for a SVM on 100 different datasets drawn from a population of 10,000 samples and compared against the true performance measured on the whole 10,000 sample population. The population was drawn from a multivariate normal distribution chosen such that the population AROC was 0.9. Each sample consisted of 1000 dimensions. Five-fold cross-validation was evaluated, with and without 10 repeats. In the case of the 10 repeats, the computational cost equalled that of the bootstrap as 50 bootstrap iterations were used. The results suggest that the .632 estimator has the most bias, but least variance. The ϵ-0 estimator is similar to the cross-validation results, however appears to have lower variance. Repeating the cross-validation 10 times reduced the variance slightly, but not to the level of the ϵ-0. Given these results, it would seem the cross-validation is a better estimator if computational resources are limited or the least bias is desired, but given sufficient computational time the bootstrap appears to be the better

estimator, though the choice among the various bootstrap approaches (ϵ-0, .632, and .632+) would depend on the amount of bias one is willing to accept for a reduction in variance.

2.7 Model Selection

What has yet to be discussed is how to select a good model given a set of different models. This is an important question as many different models can be produced using the feature selection techniques and by varying the hyperparameters (e.g., λ for minimisers of the regularised risk – see Definition 2.19). How to select a good model is not clear as it is confounded by overfitting issues, especially when $m \gg n$; one cannot simply select the best model by minimising the error on the training data as this is likely to select an overfit model. The *Akaike information criterion* (AIC) attempts to correct for this overfitting bias by penalising by the size of the model. The best model is the model that minimises the AIC.

Definition 2.31 (Akaike information criterion, (Akaike, 1974)). *The* AIC *for a given model f_θ induced using hyperparameters θ on training data $\mathcal{X}$ is given by*

$$\text{AIC} = -\ln \text{lik}(\mathcal{X}, f_\theta) + C(f_\theta),$$

where lik *is the likelihood and C is the degrees of freedom of the prediction rule f_θ.*

In the case of linear models where $f(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0$ with normal noise assumptions, lik $= 1/\text{RSS}$ and $C(f) = ||\boldsymbol{\beta}||_0 + 1$.

The *Bayesian information criterion* (BIC) resembles the AIC as it also penalises the log-likelihood, but is derived using Bayesian principles from a set of prior assumptions. The full derivation is not provided here, but can be found from various sources (Hastie et al., 2001; Schwarz, 1978)

Definition 2.32 (Bayesian information criterion (Hastie et al., 2001; Schwarz, 1978)). *The* BIC *for a given model f_θ induced using hyperparameters θ on training data $\mathcal{X}$ is given by*

$$\text{BIC} = -2 \ln \text{lik}(\mathcal{X}, f_\theta) + C(f_\theta) \ln n,$$

where lik *is the likelihood, C is the degrees of freedom of the prediction rule f_θ, and n is the number of samples used for training.*

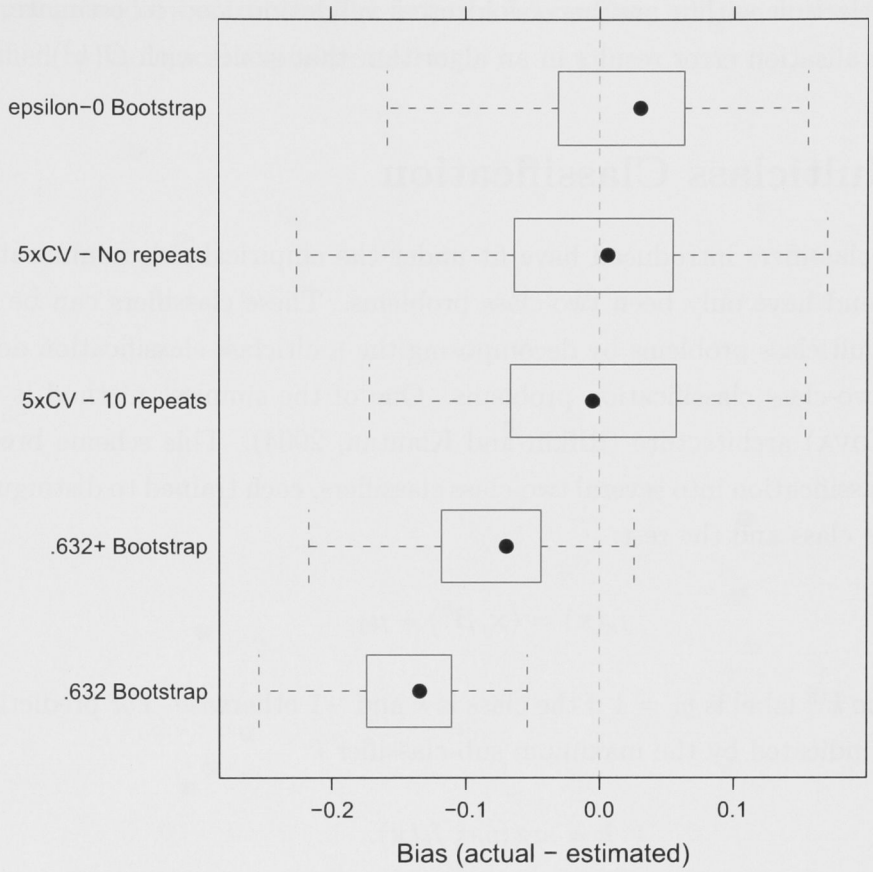


Figure 2.4: A comparison of various resampling procedures for estimating AROC. Boxplots show the distribution of AROC estimations over 100 different datasets containing 1000 dimensions and 100 samples. Boxes indicate lower and upper quartiles, and the dot indicates the median. Whiskers show min/max values.

Here one can see the BIC and AIC are very similar, and only differ with scaling of the penalty term. As the BIC multiplies by $\ln n$, the BIC penalises complex models more heavily when $n > e^2$.

Though the BIC prefers smaller models, when $n \ll m$ it still tends towards overly complex models (Broman, 2002). An alternative approach to model selection is to estimate the generalisation error – using one of the techniques presented in Section 2.6 – and choose the model which minimises the generalisation error. This approach assumes nothing about the model or its complexity, and assumes better models generalise better. Though not accounting for complexity directly, this approach does tend to select simpler models as overly complex models will not generalise as well. The disadvantages of this method is the computational

cost, and the smaller training set size. For example, using k-fold cross-validation for model selection within another k-fold cross-validation loop to estimate the overall generalisation error results in an algorithm that scales with $O(k^2)$.

2.8 Multiclass Classification

So far, the classifiers introduced have fit under the empirical risk minimisation framework and have only been two-class problems. These classifiers can be extended to multiclass problems by decomposing the multiclass classification down to several two-class classification problems. One of the simplest method is the *one-vs-all* (OVA) architecture (Rifkin and Klautau, 2004). This scheme breaks down the classification into several two-class classifiers, each trained to distinguish between one class and the rest:

$$f_k(\mathbf{x}) = \left\langle \mathbf{x}, \boldsymbol{\beta}^k \right\rangle + \mu_0,$$

assuming the k^{th} label is $y_k = 1$ if the class is k and -1 otherwise. For prediction, the class is indicated by the maximum sub-classifier k:

$$\hat{k} = \arg\max_k f_k(\mathbf{x}).$$

As one classifier is trained per class, the architecture scales linearly with $O(n_c)$ where n_c is the number of distinct classes. Another similar method is the all-pairs architecture, where each sub-classifier distinguishes between a pair of classes. This method has the advantage of not requiring the individual sub-classifiers to be directly comparable, with the final class prediction determined by voting. However, it scales in polynomial time with $O(n_c^2)$.

2.9 Summary

This review has touched on the topic of statistical machine learning (SML). Many techniques for building predictors and how to evaluate the performance of predictors given some finite subset of data were presented. These techniques form the basis of everything presented in this thesis; every chapter deals with building predictors for characteristics of future samples. Feature selection is an important concept used repeatedly to provide insight into the underlying systems that generated the data through selection of relevant features. Note that a feature

considered relevant for prediction may not be biologically relevant, hence any insight gained by examining the features selected are not conclusive and must be verified by biological experiments.

Chapter 3

Quantitative Trait Loci Mapping

A *quantitative trait* – or *phenotype* – is any continuous characteristic of an organism that can be physically measured (e.g., plant height). A *quantitative trait locus* is a particular gene which contributes to variation of a trait. The goal in *Quantitative trait loci* (QTL) mapping is to estimate the number of QTL, the strength of each QTL, and the location of the QTL in the genome. There may be interactions between QTL, which is known as *epistasis*, however most QTL mapping methods assume there is no epistasis for simplicity.

Knowledge of these regions can aid in the understanding of the underlying biological system, and can ultimately help in the breeding of better crops through more informed crosses and early identification of promising lines. QTL mapping is also used in areas other than plant genomics such as human cancer genomics; an example application in this field is the detection of cancer-related QTL that give a genetic predisposition towards carcinogenesis within members of a family. With the emergence of the new high-density single nucleotide polymorphism (SNP) arrays for humans, QTL methods are becoming applicable to more problems.

Previous approaches to QTL profiling have been heavily based on traditional statistical techniques that are more concerned with hypothesis testing and small models rather than generalisation ability. Many of the techniques presented in the review, Section 3.1, rely on either a single QTL assumption, or by initially attempting to fit a single-QTL model and then expanding the model size by searching for additional covariates. While this mode of operation has served the mapping community well, with the rapid increase of marker quantity and density due to emerging cost-efficient technologies such as microarrays and increased computing resources, modern data-mining techniques and machine learning methods are starting to play a larger role. Furthermore, the traditional forward selec-

tion model building procedure has the disadvantage of considering each feature independently and not holistically.

This chapter examines the problem of QTL mapping approached from the perspective of machine learning. The focus of the mapping techniques is on estimating and measuring the generalisation error rather than hypothesis testing; this focus perhaps leads to more conservative estimates of putative QTL effects, but has the advantage of being supportable by the available data. Furthermore, the problem is approached by analysing the whole genome rather than individual features.

3.1 A Review of QTL Mapping

This review covers the a range QTL mapping techniques from classical methods to more recent propositions (Alberts et al., 2002; Balding et al., 2001; Broman, 2001; Doerge et al., 1997; Kearsey and Hyne, 1994; Knapp et al., 1990; Tanksley, 1993).

3.1.1 Genetics and Recombination Models

Living organisms are comprised of one or more cells each containing *deoxyribonucleic acid* (DNA). DNA is a polymer of nucleotides connected by a phosphate-deoxyribose chain comprising of four bases: Adenine (A), Guanine (G), Thymine (T), and Cytosine (C). Nucleotides form base pairs that are arranged in two strands twisted together to form a *double helix*. Nucleotide pairs form only between specific bases – A only pairs with T, and G only pairs with C. Such specific binding means that a full double helix can be reconstructed exactly from either of the two nucleotide strands. Replication of the double helix from a single strand is accomplished by the DNA *polymerase* enzyme, which binds each nucleotide in the strand with its opposite pair. During cell division, the double helix of the original cell is separated into the two individual strands. The DNA polymerase enzyme then replicates two identical double helices. These helices are then segregated into two different cells by the process of *mitosis*.

In eukaryotic organisms, the genome – i.e., the entire DNA sequence – is divided into different chromosomes containing specific sequences of DNA. The number and type of chromosomes are characteristic of the organism's species and sex. Human beings, for example, have two sets of 23 chromosomes, one set received from

each parent. A gene is a *consecutive* functional sequence of base pairs in a chromosome. Genes were initially considered "units of inheritance," however recent results demonstrated the presence of RNA-based inheritance in mammals (Rassoulzadegan et al., 2006), and RNA-directed overwriting of DNA in plants (Lolle et al., 2005). Thus, note that the concept of a gene is still changing, with a recent definition being "... a union of genomic sequences encoding a coherent set of potentially overlapping functional products" (Gerstein et al., 2007).

Diploid sexually reproducing organisms have two chromosome sets, one from each parent, and *polyploid* organisms contain more than two chromosome sets. Diploid organisms are common among animals, however there are many polyploid organisms among plants. Polyploid organisms are difficult to model, thus the focus will be on diploid organisms for this review.

An *allele* is a particular sequence of a gene. The standard notation is to label each allele by a capital letter, for example A or B. A *genotype* is the combination of alleles at a specific gene, for example AA or AB for a diploid organism. In a *pure line*, the alleles are equal at all loci. For example, a diploid pure line may have genotype AA, and another different pure line may have BB, at every location of the genome.

In sexual reproduction, the process of meiosis forms *gametes* – haploid cells containing a single chromosome set – which are then combined during fertilisation to produce the offspring genotype. Gametes are formed following chromosome replication, genetic recombination (crossover), and cell division. During crossover, the chromosome pairs exchange genetic material. Each point along the chromosome has a small probability of being a crossover site, and the *frequency of recombination* between two genes on a chromosome is proportional to the distance between them. The gametes of a pure parent with alleles AA is A at all loci, as crossing over between chromosomal pairs does not introduce any genotypic variations.

When two pure parents with genotypes AA and BB are crossed, the first generation is called the *first filial generation* or F_1 progeny. This generation has only one possible genotype, AB at all loci, and so the entire progeny will have an identical expression of all traits (ignoring non-genetic variations). By inbreeding another generation from the F_1 progeny, creating the F_2 progeny, the possible genotype at loci expands to AA, AB, and BB, with frequencies of $\frac{1}{4}$, $\frac{1}{2}$, and $\frac{1}{4}$.

A common experiment design is the *back cross* (BC) of inbred lines. For this experiment, two pure parent lines are crossed producing the F_1 progeny, which is

then crossed with one of the parents producing the BC progeny. The BC progeny has the favourable characteristic of only having two possible genotypes that occur with equal probability – for example, individuals in the offspring of the BC progeny produced by crossing the F_1 progeny with the AA parent have AA and AB as possible genotypes with equal probability. Figure 3.1 illustrates this example.

Although one can sequence an entire genome, current technology is prohibitively expensive for tasks such as sequencing crops. Cheaper technology is available, but with these one cannot sequence the entire genome and can only determine the inheritance at certain loci. These loci are called *markers* and the density of markers varies with the technology used. The QTL mapping techniques reviewed in later sections usually assume the availability of a *marker map* that provides the genetic distances between markers.

Genetic distance is measured in two ways: the recombination rate, and in Morgans (M) or centiMorgans (cM). The recombination rate between two loci is the probability of an exchange during meiosis leading to gamete formation. As the recombination rate is a probability, it is not additive. The genetic distance in Morgans is the expected number of crossovers between two loci. Unlike the recombination rate, genetic distances in Morgans are additive and satisfy the requirements of a metric. Because of this, marker maps are frequently specified in centimorgans. The genetic distance and physical distance (the number of base pairs between two points) do not have a fixed relationship; some regions of the genome are more predisposed to crossover events than other regions.

The Haldane and Kosambi *mapping functions* are commonly used to convert between Morgans and the recombination rate. Haldane's mapping function is the simplest, which assumes there is no interference[1]. Haldane used a Poisson process defined by the density function $P(k) := \frac{e^{-m}m^k}{k!}$, where m is the expected number of crossovers (distance in Morgans) and k is the number of crossovers. Using a Poisson process is justifiable as the crossover events are random, largely independent, and occur with low probability. The probability of the genotype being different between two loci is equal to the probability of an odd number of crossovers occurring.

Proposition 3.1 (Haldane's mapping function). *The probability of the genotype differing between two loci is* $\frac{1}{2}(1 - e^{-2m})$, *assuming a Poisson process and no interference.*

[1] a change in the probability of a crossover caused by a nearby crossover

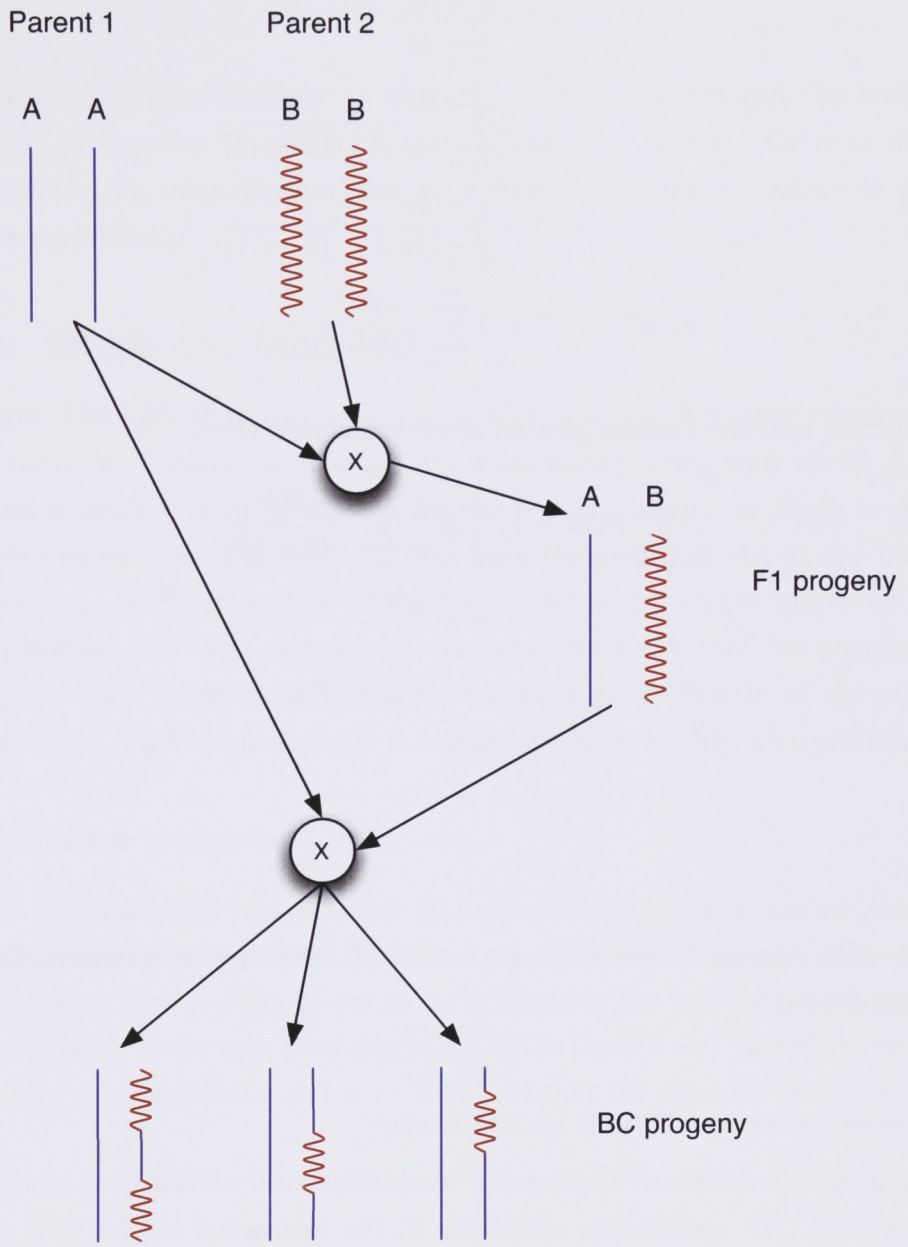


Figure 3.1: Backcross experiment illustration. Two pure parents are crossed to form the F1 progeny. A cross between the F1 progeny and a pure parent then produces the backcross progeny. The lines represent a chromosome pair that alters slightly in DNA sequence between the two "pure" parents (indicated as waved red versus straight blue lines). The recombination process in the F1 progeny produce gametes with "reshuffled" chromosome fragments.

Proof. From the definition of e^m:

$$\begin{aligned} e^m &= \sum_{k=0}^{\infty} \frac{m^k}{k!} \\ &= \sum_{k=0}^{\infty} \frac{m^{2k}}{(2k)!} + \frac{m^{2k+1}}{(2k+1)!} \\ e^{-m} &= \sum_{k=0}^{\infty} \frac{m^{2k}}{(2k)!} - \frac{m^{2k+1}}{(2k+1)!} \\ \Rightarrow e^m - e^{-m} &= 2\sum_{k=0}^{\infty} \frac{m^{2k+1}}{(2k+1)!}. \end{aligned}$$

Substituting into the Poisson process density function gives

$$\begin{aligned} \sum_{k=0}^{\infty} P(2k+1) &= e^{-m} \sum_{k=0}^{\infty} \frac{m^{2k+1}}{(2k+1)!} \\ &= e^{-m} \frac{e^m - e^{-m}}{2} \\ &= \frac{1}{2}(1 - e^{-2m}). \end{aligned}$$

□

The other commonly used mapping function is the empirically derived Kosambi mapping function

$$\frac{1}{2} \frac{e^{4m} - 1}{e^{4m} + 1},$$

which does not assume there is no interference.

If one were to work at the marker resolution level, an additive model that assumes each QTL contributes additively to the expression of the trait can be used; this is a linear model as introduced in the statistical machine learning section (Chapter 2). Assume a BC experiment with 2 possible genotypes, AA and AB occurring with equal probability. Let the data be

$$\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \{-1, 1\}^m \times \mathbb{R},$$

where y_i is the phenotype for an individual, $\mathbf{x}_i$ is an m-vector specifying the genotype of an individual, with 1 coding for AA and -1 coding for AB, and

$\boldsymbol{\beta} = (\beta_i)^m \in \mathbb{R}^m$. The prediction function is then

$$f(\mathbf{x}) := \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0.$$

The parameter β_i specifies the i^{th} marker's contribution towards the trait variance, i.e., $\boldsymbol{\beta}$ specifies the strength and direction of influence for each marker. The task of QTL mapping can now be defined as finding a reasonable pair of parameters $\boldsymbol{\beta}$ and μ.

3.1.2 Single QTL Models

To begin, methods that assume there is only a single QTL (per chromosome) are introduced. Consider a BC generated from pure parents with alleles AA and AB, and a single QTL at position x for the trait y. Let $\mu_A := E[y|x = AA]$ be the mean phenotype of individuals that have the genotype AA at the putative QTL, and $\mu_B := E[y|x = AB]$ be the mean phenotypes of the individuals that have genotype AB. The strength of the QTL effect can then be estimated as $\Delta := \mu_A - \mu_B$. Unfortunately, this cannot be applied directly as the putative QTL genotype, x, is unknown as it is unlikely to align exactly with a marker.

Single marker regression

Instead of considering the putative QTL directly, let x be a marker that is a recombination distance r away from the QTL. Figure 3.2 illustrates this case of a chromosome with a single putative QTL. Suppose the marker genotype is AA. The probability of the QTL genotype being AA is then $(1-r)$, and the probability of the QTL genotype being AB is r. It follows that the *expected phenotype* is

$$E[y|x = AA, r] = (1 - r)\mu_A + r\mu_B = \mu_A - r\Delta.$$

Similarly, if the marker genotype is AB then the expected phenotype is

$$E[y|x = AB, r] = (1 - r)\mu_B + r\mu_A = \mu_B + r\Delta.$$

The difference between the two expectations is

$$\begin{aligned} \beta &:= \mu_A - r\Delta - \mu_B - r\Delta \\ &= (1 - 2r)\Delta \end{aligned} \tag{3.1}$$

Therefore, a non-zero β calculated for the marker indicates a link with the QTL, with decreasing power as r increases.

QTL can therefore be detected by estimating β_j for each marker j, and applying a threshold to the results to determine significance. For instance, given a dataset

$$\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \subset \{-1, 1\}^m \times \mathbb{R},$$

where $\{1, -1\}$ indicates the genotypes AA an AB respectively, then using a single marker linear model

$$y = x^{(j)}\beta_j + b_j$$

we have

$$\beta_j = E[y|x^{(j)} = 1] - E[y|x^{(j)} = -1] \text{ and } b_j = E[y|x^{(j)} = -1],$$

and the estimated variance explained for the j^{th} marker is $\text{cor}(\mathbf{x}^{(j)}, \mathbf{y})^2$. The advantage of this method is simplicity and computational efficiency, but it has several disadvantages. First, using this method does not produce separate values for the QTL location r and the QTL effect Δ. Another problem is that the power for QTL detection depends on the density of the markers. With a low marker density, the QTL may be far from all markers, hence the measured effect β_j at markers j will be small. This can be seen by observing that as $r \to \frac{1}{2}$, the measured effect $\beta_i \to 0$. However, both these disadvantages are lessened by the density available with current genotyping techniques.

LOD scores

To resolve the problems of marker resolution, the *logarithm of difference* (LOD) scores were proposed. LOD scores explicitly model the location r of the QTL from the marker. The model is then solved for both the location and effect using maximum likelihood (ML) techniques.

Consider the same BC experiment as previously. Let x be a marker with genotype $x \in \{AA, AB\}$. Assume the phenotypes due to the QTL are normally distributed and *homoscedastic* – i.e., the variance of the AA and AB groups are the same. Let the phenotypes of the observations with genotype AA at the QTL be $y_{AA} \sim N(\mu_A, \sigma)$, and the phenotypes of the observations with genotype AB at the QTL be $y_{AB} \sim N(\mu_B, \sigma)$, where $N(\mu, \sigma)$ denotes the normal distribution with mean μ and standard deviation σ. The phenotype distribution due to the

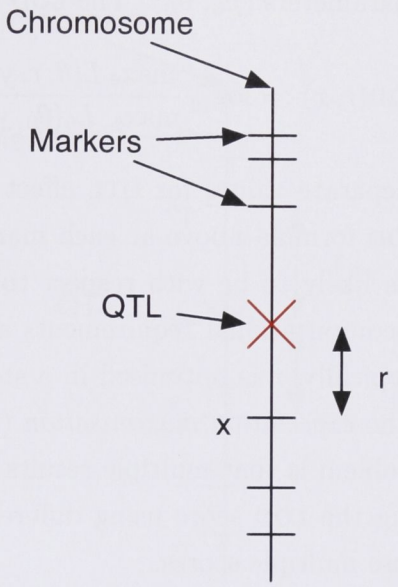


Figure 3.2: Illustration of a single putative QTL on a chromosome positioned a recombination distance r away from the marker x.

QTL is then a mixture of two Gaussian distributions with a probability density function of

$$f(y|\theta, r, x) = \begin{cases} \frac{1-r}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_A)^2}{2\sigma^2}) + \frac{r}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_B)^2}{2\sigma^2}) & \text{if } x = \text{AA} \\ \frac{r}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_A)^2}{2\sigma^2}) + \frac{1-r}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_B)^2}{2\sigma^2}) & \text{if } x = \text{AB} \end{cases},$$

where $\theta = (\mu_A, \mu_B, \sigma)$ is a vector of parameters. Let $\mathbf{y} \in \mathbb{R}^n$, where y_i is the phenotype for observation i and n is the number of observations. The likelihood is then

$$L(\theta, r, \mathbf{y}, x) := \prod_i^n f(y_i|\theta, r, x).$$

The LOD score is a likelihood ratio test between this model and the "no-QTL" null-hypothesis. If there is no QTL, all the phenotype values are assumed to be normally distributed with mean μ_0 and variance σ_0^2. Therefore, the null-hypothesis density and likelihood functions are:

$$f_0(y, \theta_0) := \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_0)^2}{2\sigma_0^2}) \text{ and}$$
$$L_0(\theta_0, \mathbf{y}) = \prod_i f_0(y_i, \theta_0),$$

where θ_0 is the tuple of parameters (μ_0, σ_0). The LOD score is then defined as

$$\text{LOD}(r, x) := \log_{10} \frac{\max_\theta L(\theta, r, \mathbf{y})}{\max_{\theta_0} L_0(\theta_0, \mathbf{y})}.$$

LOD scores provide separate values for QTL effect and location based on a marker. Applying the LOD formula above at each marker for various r will determine where the QTL is likely to be with respect to the marker position and genotype. However, the computational requirements are much greater than for single marker analysis; typically, r is optimised in a stepwise fashion with θ optimised for each r using the *expectation maximisation* (EM) algorithm (Dempster et al., 1977). Another problem is that multiple results for the same position are obtained when calculating the LOD score using different markers, and it is not clear how to combine these multiple scores.

3.1.3 Interval Mapping

Lander and Botstein (1989) proposed a technique known as *interval mapping* that can overcome the multiple results problem when using LOD scores. Interval mapping uses two flanking markers to estimate the strength of a putative QTL– see Figure 3.3. The following sections discuss the extension of LOD scores to interval mapping, and also introduces other flanking marker methods.

LOD scores

The extension of LOD scores to interval mapping is relatively simple. Again, consider the same BC experiment and let x_l and x_r be two markers flanking a putative QTL with genotypes $x_l, x_r \in \{AA, AB\}$. As before, assume the phenotypes are normally distributed and homoscedastic with respect to the genotype groups:

$$y_{AA} \sim N(\mu_A, \sigma^2) \text{ and } y_{AB} \sim N(\mu_B, \sigma^2).$$

Let α be the probability of the genotype at the putative QTL being AA. The density function can now be rewritten as:

$$f(y|\theta, \alpha) = \frac{\alpha}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_A)^2}{2\sigma^2}) + \frac{1-\alpha}{\sigma\sqrt{2\pi}} \exp(-\frac{(y-\mu_B)^2}{2\sigma^2})$$

Let r_l be the recombination distance between the QTL and the left marker, r_r be the recombination distance between the QTL and the right marker, and r be

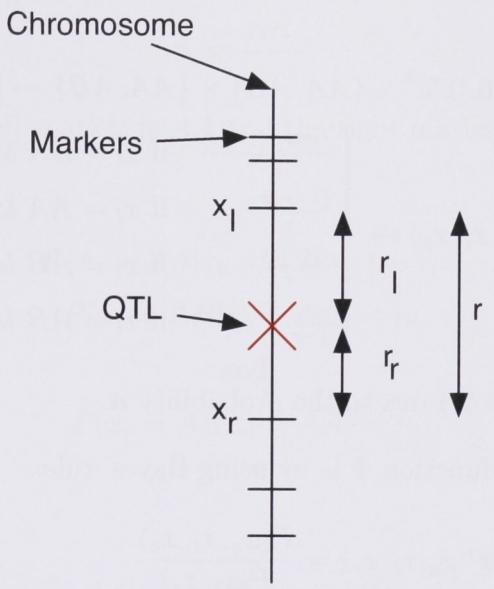


Figure 3.3: An illustration of interval mapping. A putative QTL is shown with the relevant recombination distances from the flanking markers. Note that the recombination rates are not additive ($r_l + r_r \neq r$).

the recombination distance between markers (Figure 3.3). Recall that the recombination rates are the probabilities of the genotype differing between two points and are not additive ($r_l + r_r \neq r$) due to the possibility of multiple crossovers of even order (e.g, double crossovers). The likelihood functions and LOD score function given a set of observations now follow easily:

$$L(\theta, \mathbf{r}, \mathbf{y}, x_l, x_r) = \prod_i^n f(y_i|\theta, \Phi(\mathbf{r}, x_l, x_r))$$

$$\text{LOD}(\mathbf{r}, \mathbf{y}, x_l, x_r) = \log_{10} \frac{\max_\theta L(\theta, \mathbf{r}, \mathbf{y}, x_l, x_r)}{\max_{\theta_0} L_0(\theta_0, \mathbf{y})},$$

where $\mathbf{y} = (y_i) \in \mathbb{R}^n$ is the phenotypes of the individuals, $\mathbf{r} = (r_l, r_r, r) \in [0, 0.5]^3$ is a tuple denoting the position of the QTL based on the flanking markers (see

Figure 3.3), and

$$\Phi\colon [0,0.5]^3 \times \{AA, AB\} \times \{AA, AB\} \to [0,1] \tag{3.2}$$

$$((r_l, r_r, r), x_l, x_r) \mapsto \begin{cases} \frac{(1-r_l)(1-r_r)}{1-r} & \text{if } x_l = AA \ \&\ x_r = AA \\ \frac{(1-r_l)r_r}{r} & \text{if } x_l = AA \ \&\ x_r = AB \\ \frac{r_l(1-r_r)}{r} & \text{if } x_l = AB \ \&\ x_r = AA \\ \frac{r_l r_r}{1-r} & \text{if } x_l = AB \ \&\ x_r = AB \end{cases} \tag{3.3}$$

maps the recombination rates to the probability α.

The derivation of function Φ is by using Bayes' rule:

$$\begin{aligned} P(x_q|x_l, x_r) &= \frac{P(x_q, x_l, x_r)}{P(x_l, x_r)} \\ &= \frac{P(x_r|x_q, x_l)P(x_q, x_l)}{P(x_l, x_r)} \\ &= \frac{P(x_r|x_q)P(x_q, x_l)}{P(x_l, x_r)} \\ &= \frac{P(x_r|x_q)P(x_q|x_l)P(x_l)}{P(x_r|x_l)P(x_l)} \\ &= \frac{P(x_q|x_l)P(x_r|x_q)}{P(x_r|x_l)}, \end{aligned}$$

where x_q, x_l, and x_r are the genotypes of the putative QTL, and the left and right flanking markers.

First, assume both flanking markers have genotype AA. Then,

$$P(x_q = AA|x_l = AA) = 1 - r_l,$$
$$P(x_r = AA|x_q = AA) = 1 - r_r,$$
$$\text{and}$$
$$P(x_r = AA|x_l = AA) = 1 - r.$$

It follows that

$$\alpha = P(x_q = AA|x_l = AA, x_r = AA) = \frac{(1 - r_l)(1 - r_r)}{1 - r}.$$

Similarly, if both markers are AB then the probability of the QTL genotype being

AA is

$$\alpha = \frac{r_l r_r}{1 - r}.$$

Now assume the left marker is AA and the right marker is AB. The probabilities in this case are

$$P(x_q = AA|x_l = AA) = 1 - r_l,$$
$$P(x_r = AB|x_q = AA) = r_r,$$
and
$$P(x_r = AB|x_l = AA) = r.$$

It follows that

$$\alpha = P(x_q = AA|x_l = AA, x_r = AB) = \frac{(1 - r_l)r_r}{r}.$$

Similarly, if the left marker is AB and the right marker is AA, then the probability of the QTL genotype being AA is

$$\alpha = \frac{r_l(1 - r_r)}{r}.$$

The remaining definition of Φ follows easily.

This LOD score for an entire chromosome is typically calculated by stepping through the chromosome positions with a small step resolution (1cM or so). At each step, θ is maximised using EM. This method provides a single measure of QTL effect at any location, avoiding the multiple results problem encountered during single marker LOD mapping. Figure 3.4 shows an example LOD curve generated using EM fitting.

3.1.4 Regression Methods

As interval mapping requires many repeated applications of EM, the computational cost is high. Regression methods do not have the repeated application of EM, and thereby have reduced computational cost. Several regression methods (Haley and Knott, 1992; Kearsey and Hyne, 1994; Knapp et al., 1990) have been proposed that offer precision similar to LOD scores, yet run many times faster.

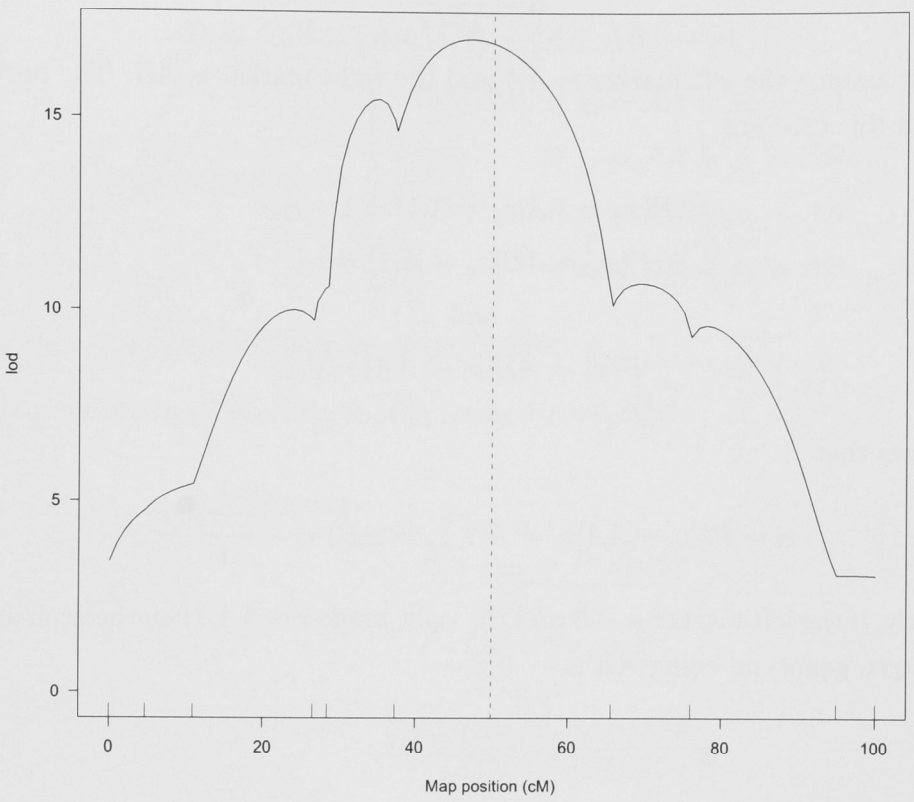


Figure 3.4: An example LOD curve for simulated data. The ticks along the x-axis indicate the marker locations, and the y-axis indicates the LOD score. A single QTL at 50cM (marked as red line) was simulated.

Regression mapping

Haley and Knott (1992) proposed a mapping method known as *regression mapping*. Consider the same configuration as in interval mapping (Figure 3.3). The expected phenotype value is

$$E[y|x_l, x_r] = \mu_B + (\mu_A - \mu_B)\Phi(\mathbf{r}, x_l, x_r)$$

where x_l and x_r are the flanking marker genotypes, $\mathbf{r} = (r_l, r_r, r)$ are the recombination fractions, and Φ is as in Equation 3.3. The parameters μ_A and μ_B can then be found easily by minimising the residual sum of squares (RSS, see Definition 2.4). LOD scores can be calculated from the RSS under the assumption that the RSS is normally distributed, which is true given an adequate number of samples due to the central limit theorem. Let $\text{RSS} := \sum_i (y_i - \hat{y})^2$ be the residual sum of squares

of the regressed Haley-Knott model (μ_A and μ_B), $\text{RSS}_0 := \sum_i (y_i - \bar{y})^2$ be the residual sum of squares for the "no-QTL" null-hypothesis, $\bar{y} := \frac{1}{n}\sum_{i=1}^{n} y_i$, and n be the number of observations. The LOD score is then

$$\text{LOD}(\text{RSS}, \text{RSS}_0, n) := \frac{n}{2}\log_{10}(\frac{\text{RSS}_0}{\text{RSS}}).$$

This method is substantially faster than the original interval LOD score method as the repeated applications of EM are avoided. Haley and Knott (1992) demonstrated that this regression method's precision is similar to the original LOD score method.

Marker regression

Kearsey and Hyne (1994) proposed another simple regression method called *marker regression*. Unlike the Haley-Knott method, it does not use flanking markers, but instead incorporates the recombination fractions for all markers simultaneously. Recall Equation 3.1 showing that the difference in the marker means is $\beta_i = (1 - 2r_i)\Delta$, where Δ is the phenotype difference between genotype groups at a putative QTL, β_i is the difference at a marker i, and r_i is the recombination fraction between the putative QTL and the marker. Instead of regressing directly $\mathbf{y} \sim \mathbf{x}$, Kearsey and Hyne (1994) proposed regressing

$$\beta_{ij} := \text{cor}(\mathbf{x}^{(j)}, \mathbf{y}) \sim r_{ij}$$

where r_{ij} is the recombination fraction between the j^{th} marker and the i^{th} putative QTL position, and $\mathbf{x}^{(j)}$ is the vector of samples associated with the j^{th} marker. Doing so gives $\Delta_i = \text{cor}(\mathbf{r}_i, \boldsymbol{\beta}_i)$ where the variance explained of the observed marker variances for each putative QTL j is given by Δ_j^2. Like the Haley-Knott method, this regression method is computationally faster then the LOD score EM method.

3.1.5 Multiple QTL Models

Although the single QTL models can be effective for simple organisms and traits, when multiple QTL are present the detection power is less due to more variance in the phenotypes (Knapp, 1991; Lander and Botstein, 1989). The assumption of no epistatic effects also serves to lower the detection power. As many organisms have multiple QTL, the loss in detection power can be quite substantial. Current

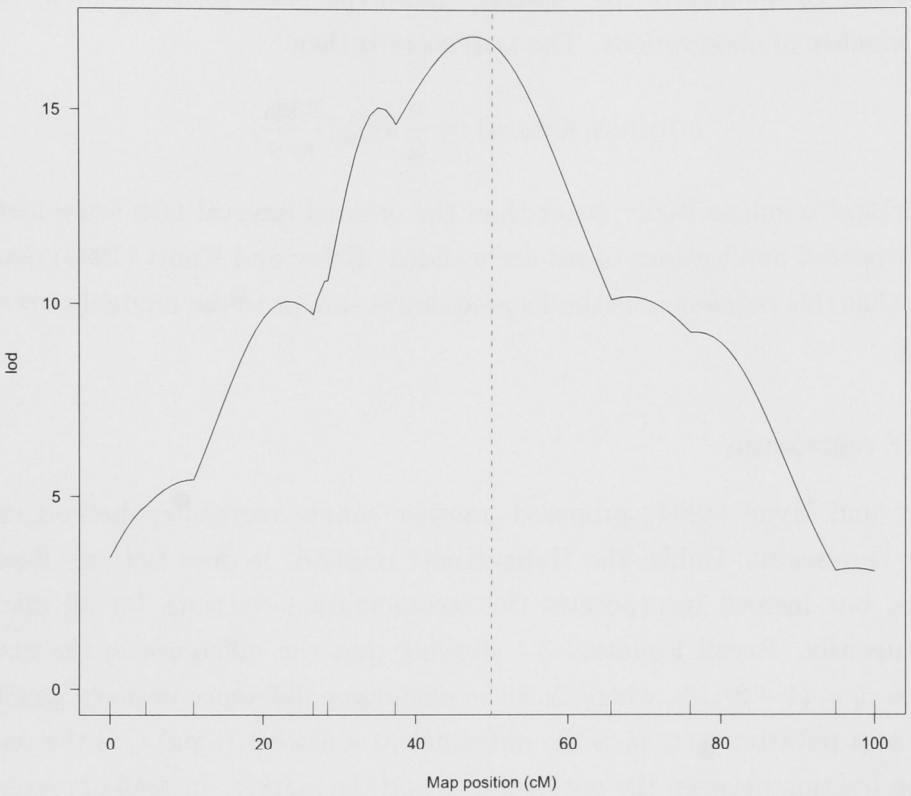


Figure 3.5: An example LOD curve for simulated data using regression mapping. The ticks along the x-axis indicate the marker locations, and the y-axis indicates the LOD score. A single QTL at 50cM (marked as red line) was simulated.

QTL research has therefore concentrated on modelling multiple QTL.

Interval mapping and derivatives

Although interval mapping is generally considered a single QTL model, Lander and Botstein (1989) did propose a multiple QTL forward selection procedure. It was suggested that upon the observation of multiple peaks in a genome profile, to fix the position of the major peak as a QTL in the model, and then calculate a second LOD curve on the residual to determine the other QTL locations. Very broad flat peaks are handled the same way, as here the assumption is that two closely located QTL are creating a wide region of significance. This, in essence, is a heuristic forward selection procedure.

However, this procedure can suffer from a "ghost QTL" effect and has been criticised in the literature (Broman, 2001; Haley and Knott, 1992). In essence,

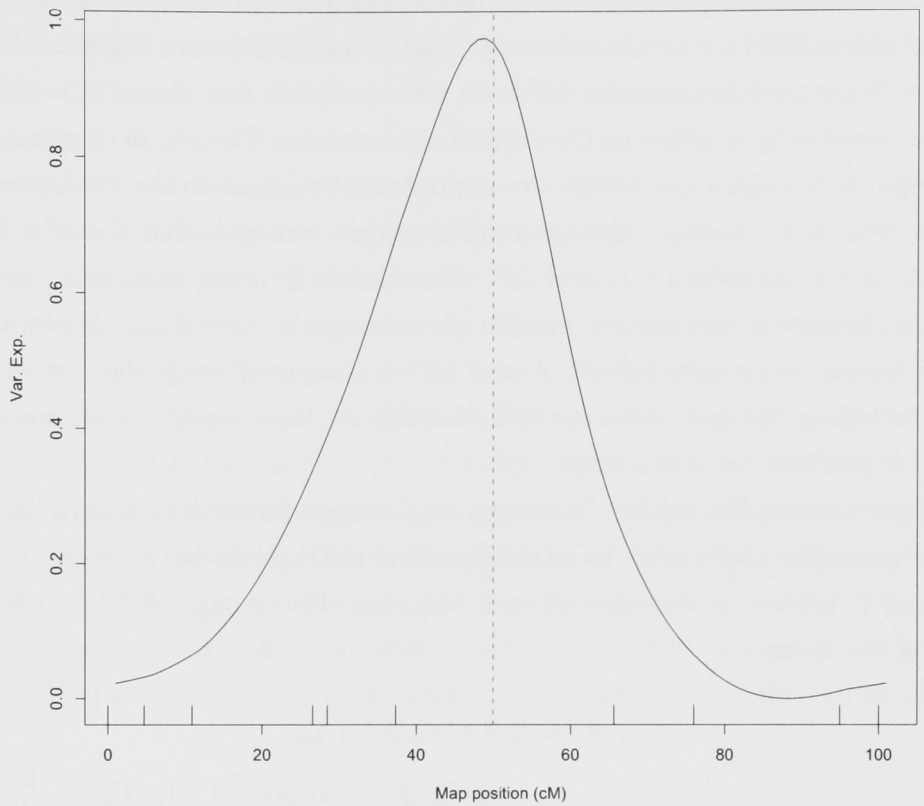


Figure 3.6: An example variance explained curve for simulated data using marker regression. A single QTL at 50cM (marked as red line) was simulated. Note: the variance explained here is not the percentage of the phenotype variance, but rather the percentage of the observed marker variances.

when two linked QTL exist and the marker resolution is not sufficient, the LOD curve produces the major peak between the two QTL. This incorrect peak is called a "ghost QTL" as in reality there is no QTL at that location.

A method combining multiple regression and interval mapping was proposed by Jansen (1993) and Zeng (1994) under the names of *multiple* QTL *models* (MQM) and *composite interval mapping* (CIM). These methods fit a model of the form

$$E[y|\mathbf{x}] = \hat{x}\hat{\beta} + \sum_{i \in I} x_i \beta_i \tag{3.4}$$

where $\hat{x}$ is the genotype at a putative QTL, I is a subset of marker indices, and x^i is the genotype of the i^{th} marker. The second term incorporates residual variances to increase the statistical power of detecting correctly the putative

QTL $\hat{x}$. Jansen (1993) proposed commencing by finding a small initial set S that sufficiently models the phenotype using recursive feature elimination and the AIC (see Definition 2.31) for model selection. For each chromosome, Equation 3.4 is fitted at every putative location for both index subsets $I = S$ and $I = S \setminus S'$, where S' is the set of markers on the current chromosome. The AIC of these models are compared to determine if there are multiple QTL present on the chromosome; if multiple QTL are present, the assumption is that a model with $I = S$ will be selected over a model with $I = S \setminus S'$. If multiple QTL are indicated, further searching using RFE and the AIC on that chromosome is carried out. If not, then another model using only the set $I = S \setminus S'$ is compared using the AIC to the model including the best putative QTL location on the current chromosome to determine whether there is a QTL.

A more recent proposal is a technique combining a forward/backward search, interval mapping, and a term to model epistatic interactions between QTL. This technique is known as *multiple interval mapping* (Kao et al., 1999) and has a model of the form

$$E[y|\mathcal{X}] = X\boldsymbol{\beta} + X\omega X^* + \mu_0$$

where $\boldsymbol{\beta}$ is the now familiar QTL effects vector, ω is a sparse square matrix modelling the epistatic interactions, and X is the genotypic information for a finite dataset $\mathcal{X}$ in matrix form with $\mathbf{x}_i$ being the i^{th} row. Kao et al. (1999) proposed to determine $\boldsymbol{\beta}$ and ω given a set of QTL indices S and a set of interacting QTL indices $S' \subset S$ using EM, where $\beta_i = 0$ for all $i \notin S$ and $\omega_{ij} = 0$ for all i or j not in S'. The search for an optimal set S is done using a forward/backward stepwise selection procedure and hypothesis testing. Similar hypothesis testing and stepwise selection is used to determine the set S'. This method is computationally intensive as each step requires fitting of the model using EM.

3.1.6 Significance Testing

In all the methods presented so far, the question of determining which results are significant needs to be addressed. For the single QTL case, the t-test can be used to determine statistical significance for BC progenies. If more than 2 genotypes are possible (e.g., in the F_2 progeny), then a generalised form of ANOVA and F-statistics can be used.

Determining significant LOD scores is not as simple as there is no direct ana-

lytical distribution available for the null-hypothesis. The null-distribution is dependent on many factors such as the type of cross and the density of the markers. Lander and Botstein (1989) used computer simulations to determine the null distribution of maximum LOD scores for a variety of different factors. From these simulations, empirical formulae were developed for estimating appropriate LOD thresholds. These formulae are useful for most cases, however several applications exist where the assumptions required for the formulae are not met, and therefore they cannot be applied. For these cases, Churchill and Doerge (1994) proposed determining suitable threshold values based on permutation testing. This method randomly permutes the phenotypes to remove the association between trait values and the genotype, and calculates the LOD scores on the permuted data. In essence, this permutation simulates a sample from the null-distribution. By repeating this process numerous times, an empirical estimate of the null-distribution of LOD scores is obtained. This method has the advantage of being simple to apply and applicable to all datasets and mapping procedures, but suffers from a high computational cost. Churchill and Doerge (1994) recommended using at least 1000 permutations for estimating critical values at the 95^{th} percentile.

3.1.7 Whole Genome Models

The methods discussed up to now have relied on greedy selection procedures in the search for QTL. Current technology has resulted in much higher marker density than previous eras, allowing the pursuit of new methods not previously possible. These new methods analyse the whole genome and determine the QTL effect of each marker *simultaneously*, eliminating the need for nested subset searches. These embedded methods offer potentially greater synergy between the features selected and the regression coefficients, and so one hopes such a method would lead to superior QTL detection.

Recently, Xu (2003) proposed a Bayesian[2] sparse regression method that encodes a preference for small models using the Bayesian statistical framework. The model is a linear model as before ($E[\mathbf{y}|\mathcal{X}] = X\boldsymbol{\beta} + \mu_0$), but $\boldsymbol{\beta}$ is regressed by drawing from the posterior distribution using sparse priors. Encoding this preference towards sparse models encourages the phenotype to be modelled using few markers with large effect. Xu (2003) used Markov chain Monte Carlo (MCMC) sampling with 51,000 samples to simulate the posterior distribution and

[2]An introduction of Bayesian statistics is beyond the scope of this thesis. Interested readers can find information elsewhere (Gelman et al., 2003)

determine the vector $\boldsymbol{\beta}$. A further extension was proposed by Yi et al. (2005) and named Bayesian interval mapping (BIM). This method is similar as it also involves sparse Bayesian regression and MCMC sampling, however it also attempts to model epistatic interactions. As the number of possible epistatic interactions is extremely large when considering the entire genome, a liberal constraint on the maximum number of QTL is made to reduce computational cost.

3.1.8 Summary

This section gave an introduction to genetics and QTL mapping. Important concepts such as diploid, polyploid, recombination, and the BC progeny were introduced. These concepts are important in later sections, as a BC Barley (a double haploid) dataset was studied.

Many techniques for QTL mapping were introduced, ranging from simple single QTL models to the more recent Bayesian approaches. These approaches to QTL profiling do not consider generalisation ability, which will be the focus of the methods presented herein.

3.2 QTL mapping through Recursion

This section proposes a QTL mapping technique based on several techniques: ridge regression, recursive feature elimination (RFE), and bootstrap error estimation (see Section 2.6.1). Model fitting begins with ridge regression to fit a linear model using all available features. As there are many more features than samples, a high degree of regularisation is required to ensure the fitted model generalises. Recall that once a model is obtained, it can be used to determine the least useful feature for predictive performance, and then that feature can be discarded and a new model induced; this is the operating principle of RFE. The sequence of embedded models is then used to evaluate the contribution of every marker towards the predictive performance using the bootstrap for generalisation estimation. This method was originally presented by Bedo et al. (2008).

Let $\mathcal{X} := \{(\mathbf{x}_i, y_i)\}_{i \in 1,\ldots,n} \subset \mathbb{R}^m \times \mathbb{R}$ be the dataset under analysis consisting of genotype m-vectors $\mathbf{x}_i$ and continuous phenotypes y_i. It is assumed the QTL are linear and additive and hence a linear model was used:

$$f(\mathbf{x}) := \langle \boldsymbol{\beta}, \mathbf{x} \rangle + \mu_0 \tag{3.5}$$

where $\boldsymbol{\beta} \in \mathbb{R}^m$ and $\mu_0 \in \mathbb{R}$. If the data (each $\mathbf{x}^{(i)}$ and $\mathbf{y}$) is assumed to be centred[3], this simplifies to the homogeneous case where $\mu_0 = 0$:

$$f(\mathbf{x}) := \langle \boldsymbol{\beta}, \mathbf{x} \rangle .$$

Recall that the homogeneous ridge regression solution is found by minimising the regularised risk with least squares loss and L^2 regulariser, and that the solution is given by

$$\boldsymbol{\beta} = (X^* X + \lambda Id)^{-1} X^* \mathbf{y}$$

where X is the $n \times m$ matrix with rows $\mathbf{x}_i$ and columns $\mathbf{x}^{(j)}$, Id is the identity matrix, and $\lambda > 0$ is a hyperparameter controlling the amount of regularisation (see Definition 2.21). Furthermore, recall that when applying RFE to linear models, the utility of a feature can be estimated by the absolute value of its regression coefficient $|\beta_i|$ (see Section 2.5) if the feature vectors are standardised. The feature with minimum utility is then discarded and a new model fitted, resulting in a sequence of models with decreasing size, i.e.,

$$\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)}, \ldots, \boldsymbol{\beta}^{(m)} \text{ such that } ||\boldsymbol{\beta}^{(k)}||_0 = ||\boldsymbol{\beta}^{(k+1)}||_0 + 1.$$

3.2.1 Estimation of Marker Importance

Now consider the task of estimating the percentage of variance explained for a particular marker $\hat{\jmath} = \arg\min_j |\beta_{jk}|$ that was removed between models $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\beta}^{(k+1)}$. Assume that an independent testing set $\mathcal{X}_{\text{test}}$ is available for unbiased estimation of model performance, and let $f_k(\mathbf{x}) = \langle \boldsymbol{\beta}^{(k)}, \mathbf{x} \rangle$ be the prediction function for model k. The contribution of feature $\hat{\jmath}$ towards predictive performance can then be estimated as the difference between the two models:

$$\text{PAVE}_{\hat{\jmath}} = r^2(\mathcal{X}_{\text{test}}, f_k) - r^2(\mathcal{X}_{\text{test}}, f_{k+1})$$

where r^2 measures the variance explained (see Definition 2.5). This measure is named the *predictive apportioned variance explained* (PAVE) as it attributes some portion of the total predictive variance explained among the selected markers. As independent data is not available, the bootstrap can be used to create independent training and testing sets $\mathcal{X}_{\text{test}} = \mathcal{X} \setminus \mathcal{X}_{\text{train}}$. Repeated bootstrapping provides many estimates of $\text{PAVE}_{\hat{\jmath}}$, with the average providing a robust estimate for the

[3] A centred random variable has an empirical mean of 0.

predictive performance of feature $\hat{\jmath}$. Algorithm 3.1 is an explicit description of the recursive algorithm described.

If a genetic map is available, the PAVE results can be plotted to create a genome profile by simply plotting the marker position vs. the estimated variance explained. Though plotting the profile directly gives an indication of the genomic areas linked with the trait, due to the additive nature of the PAVE, a better estimate of the variance explained can be obtained by applying a small sized smoothing window. Strongly correlated features (markers in strong linkage disequilibrium) can result in a "spreading of mass" among the highly correlated group due to the order of elimination varying slightly across the bootstrap. By using a summing window the profile is smoothed, and the estimates of QTL effects improved.

3.2.2 Optimisations

The full RFE procedure described in Algorithm 3.1 is computationally intensive due to the repeated bootstrapping and model induction. Let p be the number of bootstrap iterations. As there are m models for each bootstrap iteration, the procedure scales in linear time with $O(pm)$. This can be improved by increasing the number of features pruned at each iteration; instead of discarding only 1 feature, the worst 10% are discarded. This results in an algorithm with log-linear running time $O(p \log m)$, however it will also result in a reduction of accuracy. To maintain accuracy and improve runtime, it is assumed that there are likely to be fewer than 100 important features. Under this assumption, the logarithmic reduction can be used when the number of features exceeds 100, and the discarding of single features resumed when 100 or fewer features are reached. This compromise still scales with $O(p \log m)$, but has increased accuracy during the final and most crucial eliminations.

When discarding multiple features, the change in variance explained between two models can be divided uniformly between the features removed. Following the notation of the previous section, let $\mathcal{R} \in 2^m$ be the indices of features removed between two consecutive models $\boldsymbol{\beta}^{(k)}$ and $\boldsymbol{\beta}^{(k+1)}$. Then, for all $j \in \mathcal{R}$

$$\text{PAVE}_j := \frac{r^2(\mathcal{X}_{\text{test}}, f_k) - r^2(\mathcal{X}_{\text{test}}, f_{k+1})}{|\mathcal{R}|}.$$

Algorithm 3.2 states the fast RFE-QTL algorithm.

3.3 QTL mapping through Regularisation

As previously discussed in Section 3.1.7, there are some sparse Bayesian whole-genome approaches to QTL mapping. In particular, Xu (2003) proposed a Bayesian approach where a normal prior centred at 0 was introduced for each coefficient to induce sparsity, and Markov chain Monte Carlo[4] (MCMC) used to simulate the posterior distribution. As such, it is similar to the relevance vector machine (Bishop and Tipping, 2003; Tipping, 2001), however Tipping (2001) derived analytical fixed point equations while the model by Xu (2003) is fitted using MCMC sampling. The *bayesian interval mapping* (BIM) method is similar and also uses MCMC (Yi et al., 2005), but incorporates epistatic effects.

The Bayesian method does result in sparsity, but the generalisation performance of the induced models was not explored. This is, in part, due to the computational cost that arises from using MCMC; it is far too computationally expensive to estimate the generalisation error using resampling techniques such as bootstrapping. Indeed, 51,000 MCMC samples were used by Xu (2003) to simulate the posterior distribution, making further resampling computationally unattractive. This lack of generalisation estimation is a disadvantage when dealing with high dimensional low sample size problems as the resubstitution error[5] may be low while the generalisation error is high, leading to false identification of putative QTL.

An alternative non-Bayesian approach to enforcing model sparsity is to choose a *sparse regulariser*. Sparse regularisers have the same effect as the sparse priors in the Bayesian approach resulting in many of the regression coefficients being set to zero while avoiding the computationally expensive MCMC sampling that prohibits resampling for generalisation estimation. This section presents such a method for detecting putative QTL.

Recall the empirical regularised risk equation (Definition 2.19) using the least squares loss:

$$R_{\text{emp}}[f] := \sum_i (y_i - f(\mathbf{x}_i))^2 + \lambda\Omega(f)$$

where $\Omega\colon \mathcal{F} \to \mathbb{R}$ is a *regulariser*, $\lambda \geq 0$ controls the regularisation strength, and f is our hypothesis

$$f(\mathbf{x}) := \langle \mathbf{x}, \boldsymbol{\beta} \rangle - \mu_0.$$

[4] A technique for simulating draws from a posterior distribution; see (Gelman et al., 2003) for more details.

[5] The error calculated on the training data is called the resubstitution error

A suitable hypothesis is found by minimising the empirical risk over the available training data.

To induce sparsity, the regulariser needs to be chosen such that $||\boldsymbol{\beta}||_0$ is small, however $\Omega(\boldsymbol{\beta}) = ||\boldsymbol{\beta}||_0$ cannot be chosen directly as minimising the regularised risk becomes an intractable combinatorial optimisation problem (Candès and Tao, 2005); however, the L^1 norm regulariser

$$\Omega(\boldsymbol{\beta}) := ||\boldsymbol{\beta}||_1 = \sum_i |\beta_i|$$

can be used as an approximation (see Section 2.5). This regulariser is convex and allows many of the entries in $\boldsymbol{\beta}$ to be zero (Candès and Tao, 2005; Donoho et al., 2005; Wainwright, 2006) with the hyperparameter λ controlling the sparsity. Using this regulariser, the minimal regularised risk problem becomes a constrained quadratic optimisation, and a solution can be found using quadratic programming. This regression method is known in other domains as the lasso (Tibshirani, 1996) and basis pursuit (Chen et al., 1998).

3.3.1 Estimation of Marker Importance

As in Section 3.2.1, the absolute value of the regression coefficients $|\beta_i|$ can be used as a measure of importance of marker i. The problem with using this measure directly is that it is independent of the actual performance of the model. For example, the importance of a marker $|\beta_i|$ could be large but the overall model could perform poorly, and using this measure directly would then result in a higher FPR. Furthermore, as this is a underdetermined system ($n < m$), the regression coefficients can vary significantly between different training sets. This lack of stability reduces the accuracy of QTL localisation and increases the FPR.

Both problems can be approached using the bootstrap resampling estimator introduced in Section 2.6. Given a bootstrap split $\mathcal{X}_{\text{train}}$ and $\mathcal{X}_{\text{test}}$, an *absolute* rather than a *relative* estimate of marker importance can be obtained by distributing the generalisation performance of the model based on the regression coefficients β_i:

$$\text{PAVE}_i(\mathcal{X}_{\text{test}}, f) := \frac{|\beta_i|}{||\boldsymbol{\beta}||_2} r^2(\mathcal{X}_{\text{test}}, f)$$

where $||\boldsymbol{\beta}||_2 = \sqrt{\sum_i \beta_i^2}$. As the estimated variance explained is apportioned among the markers, it shares the same name – *predictive apportioned variance explained* (PAVE) – as the previous RFE based method (see Section 3.2.1). Sim-

ilarly to the previous method, this bootstrap is repeated numerous times and a single estimate for PAVE_i obtained by estimating the centre of the distribution using the mean. As the aforementioned problem regarding strongly correlated features also affects this method, the summing window already proposed may be beneficial.

3.3.2 Optimal Hyperparameter Estimation

The selection of an appropriate hyperparameter can be viewed as a model selection problem: which is the "best" model out of a set of models obtained using different hyperparameters. Model selection statistics such as the BIC and AIC (see Section 2.7) penalise for model complexity, but they tend towards overly complex models in this domain (Broman, 2002). If it is assumed that over simplified or overly complex models do not generalise as well as models of appropriate complexity, the model with the minimum estimate of generalisation error can be selected. Thus, one method of choosing a suitable hyperparameter is to apply a *nested* bootstrap by further sampling of $\mathcal{X}_{\text{train}}$ with replacement to obtain secondary training and test sets $\mathcal{X}'_{\text{train}} \subset \mathcal{X}_{\text{train}}$ and $\mathcal{X}'_{\text{test}} = \mathcal{X}_{\text{train}} \setminus \mathcal{X}'_{\text{train}}$, fitting a variety of models on $\mathcal{X}'_{\text{train}}$ with different hyperparameters, and choosing the optimal regularisation hyperparameter as

$$\hat{\lambda} = \arg\min_{\lambda} \text{RSS}(\mathcal{X}'_{\text{test}}, f_\lambda),$$

where f_λ is the model built using the hyperparameter λ. Like the main bootstrap, a better estimate of generalisation performance – and hence better estimate of the optimal hyperparameter – can be obtained by repeating the nested bootstrap procedure several times. The full algorithm is outlined in Algorithm 3.3

3.3.3 Approximate Solutions

The full method outlined in the previous few sections requires significant computational time due to the nested bootstrapping and the quadratic optimisation. It is, therefore, quite tempting to use an alternative approximate solution rather than the full procedure. The first obvious optimisation is to skip the inner bootstrap employed for estimating the optimal hyperparameter. To this end, one can use the BIC previously mentioned in the hope that although the models produced will be overly large, the final bootstrapped PAVE values will be reasonably

accurate (see Algorithm 3.4).

Another optimisation is to avoid the quadratic programming step for finding a solution. A method known as stagewise regression – in essence a boosting method – is known to produce solutions close to the L^1 solution without the quadratic optimisation (Efron et al., 2004).

The termination conditions for the loop must be such that the final model is of sufficient size. This problem is similar to choosing the best λ which was discussed previously. To detect when the model has grown sufficiently, *random probes* – false features randomly generated and added to the dataset – were used. Two termination conditions were used:

1. The same feature is maximally correlated with the residual in two consecutive iterations, but with opposite signs

2. When a random probe is maximally correlated with the residual

The first condition avoids infinite looping caused by an insufficiently refined step size. The second condition terminates the loop when a known unrelated feature is selected. The more random probes added to the dataset, the lower the type-I error rate (the fraction of irrelevant features selected) but the higher the type-II error rate (the fraction of relevant features not selected). In experiments, 3 random probes with a coarse step-size of $\delta = 0.1$ was used.

3.4 Results and Discussion

In this subsection, the performance of the RFE method (Algorithm 3.2) and L^1 method (Algorithm 3.3) with its approximations (Algorithm 3.4 and Algorithm 3.5) was analysed and benchmarked against *marker regression* (MR, see Section 3.1.4) and BIM methods (Section 3.1.7). Experiments were conducted on both synthetic data, and natural data from a Steptoe/Morex barley cross.

3.4.1 Synthetic Data Analysis

The various methods were evaluated on synthetic data – a backcross experiment for 100 individuals with an additive model, generated using the *R/qtl* package Broman et al. (2003). Each individual consisted of a single "chromosome" of length 20M, with markers spaced evenly every 1cM. Twenty QTL were positioned at random marker positions with random strength. There was no interference,

and Haldane's mapping function was used to convert between genetic distance and recombination fractions. Profiles for the L^1 (including optimised variants), RFE, BIM, and MR methods were generated on this synthetic data. As the QTL are positioned precisely at markers, there is no need to use interval mapping. Consequently, BIM was restricted to analysis at marker positions only. Smoothing was applied to all methods except MR, with a 5cM averaging window applied to BIM, and a 5cM summing window applied to the remainder.

Figure 3.7 shows the profiles arising from the various methods shown in blue in separate plots, as well as the true QTL position and strength shown in purple. As expected, the MR and BIM profiles overestimate the effects of the QTL, and the RFE and L^1 methods provide significantly better estimates of QTL effects. Furthermore, the sparseness of the RFE and L^1 methods far surpasses the noisy profiles obtained with BIM and MR– markers unlinked to QTL are assigned very low values. Comparing the L^1 method against the RFE method, it is clear that the L^1 estimates are better than the RFE estimates with QTL being assigned a PAVE value very close to the true QTL strength. When comparing the L^1 approximations, L^1-BIC and *Stagewise*, against the full L^1 method, the accuracy of the strength estimation drops considerably, though the sparseness is maintained quite satisfactorily. The BIC approximation appears to have estimated slightly better the strength of the QTL, but the difference is marginal and the stagewise approximation maintains the speed advantage (see Table 3.1).

Though this set of results show that the RFE and L^1 methods perform very well, it is also a limited comparison as it is only a single synthetic dataset. To further evaluate the various methods, 100 different synthetic datasets were analysed. However, it is difficult to define a good objective measure of performance, given that the identification of a QTL may be successful, though the location slightly shifted. It is also subject to any significance thresholds that may be derived (ad-hoc or otherwise) – a QTL may be "identified" but be below the threshold. The

Table 3.1: Elapsed time for bootstrap methods on the synthetic dataset. Times are averaged over 50 bootstrap iterations. Computations were conducted on a 1.83 GHz Intel Core Duo machine with 1.5 GB of memory.

Method	Time per bootstrap (ms)
Stagewise	798.2
RFE	11867.4
L^1-BIC	15684.1
L^1	24937.3

latter can be overcome by employing a performance measure such as the AROC that is independent of any threshold one chooses for classifying regions into QTL and non-QTL categories. However, the first problem still affects the AROC significantly, and in fact may penalise sparse methods more than non-sparse methods as broad regions containing a QTL will obtain a higher AROC score than narrow regions which, though close, do not contain a QTL. Thus, a "relaxed" AROC that allows peaks to be shifted by a small margin without a penalty was used.

Recall the definition of AROC:

$$\text{AROC}(\mathcal{X}, f) := \frac{1}{|I_+||I_-|} \sum_{i \in I_+} \sum_{j \in I_-} \begin{cases} 1 & \text{if } f(\mathbf{x}_i) > f(\mathbf{x}_j) \\ 0.5 & \text{if } f(\mathbf{x}_i) = f(\mathbf{x}_j) \\ 0 & \text{otherwise} \end{cases}$$

where $I_+ = \{i | y_i = 1\}$ and $I_- = \{i | y_i = -1\}$. This can be applied for evaluating QTL profiles by allowing $f(\mathbf{x}_i)$ to be the estimated variance explained for marker $\mathbf{x}_i$ by one of the methods, and by assigning

$$y_i = \begin{cases} 1 & \text{if } i \in Q \\ -1 & \text{otherwise} \end{cases}$$

where Q is the set of indices for markers linked with QTL. The definition of the AROC can be relaxed by taking the maximum variance explained among markers within a small window around QTL instead:

$$\text{AROC}'(\mathcal{X}, f) = \frac{1}{|I_+||I_-|} \sum_{i \in I_+} \sum_{j \in I_-} \begin{cases} 1 & \text{if } \max\{f(\mathbf{x}_k) | d(k,i) \leq \tau\} > f(\mathbf{x}_j) \\ 0.5 & \text{if } \max\{f(\mathbf{x}_k) | d(k,i) \leq \tau\} = f(\mathbf{x}_j) \\ 0 & \text{otherwise} \end{cases}$$

where $d(k, i) \in \mathbb{R}$ measures the genetic distance in centimorgans between the two markers i and k and $\tau \geq 0$ is the size of the window.

Figure 3.8 shows the relaxed AROC results with a 3cM window ($\tau = 2$) for the various methods on 100 synthetic datasets summarised as a boxplot. Each dataset contained 100 individuals comprising of 1 "chromosome" of 20M length. Markers were positioned every 1cM, and 20 additive QTL were randomly positioned exactly at marker positions. For methods using bootstrapping, 200 bootstrap iterations were used. From the figure it is clear that all methods outperform MR by a significant margin and achieve high levels of performance with the lower quartile

sitting well above 0.8. Somewhat surprisingly, the stagewise method appears to perform better than all the other methods with a higher median and similar variance. RFE, L^1-BIC, L^1, and BIM appear to perform very similarly with little differences between them, though RFE appears to have a smaller variance.

To evaluate the significance of difference between the various algorithms, the Tukey-Kramer method was used. Figure 3.9 shows the pairwise differences in mean with 95% confidence intervals calculated using the Tukey-Kramer method. As expected, all the comparisons against MR are significant. Furthermore, all comparisons against stagewise are significant, indicating the stagewise method is clearly the best for both performance and running time (see Table 3.1) on this particular experiment. The performance of the L^1-BIC approximation was not significantly different from the performance of the full L^1 method or BIM, however it does perform significantly better than the RFE method. The RFE, L^1, and BIM methods do not perform significantly different from each other.

3.4.2 Natural Data Analysis

Experiments on natural data, a Steptoe/Morex cross genotyped using Diversity Arrays Technology^{TM}(DArT) (Wenzl et al., 2004), were conducted. This particular cross has been well analysed in the literature (Hayes et al., 1993), though not using the DArT genotyping technology. The genotype data consisted of 96 individuals and 351 binary markers (0 or 1). Missing genotype values were substituted with 0.5. The phenotype data consists of several traits: α-amylase, heading time, height, lodging, yield, diastatic power, protein content, and pubescence. Pubescence is a phenotype governed by a single locus, and is included mainly as a positive control as the number and contribution to the variance (100%) is known. Each phenotype was measured in up to 16 different environments. As the common genetic component across all environments is the target of interest rather than the between environment variation, the 16 measurements were reduced to one using principal component analysis (PCA); each environment was scaled to have a mean of 0 and standard deviation of 1, and then the first principal component was extracted using the singular value decomposition (SVD). A one dimensional

target was obtained by projecting along this component:

$$UDV^* = Y \text{ (using SVD)}$$
$$P = V_1$$
$$\hat{Y} = YP$$

where $Y = \{y_{ij}\}_{i=1,\ldots,n;j=1,\ldots,m} \subset \mathbb{R}$ is the measurements for one phenotype for n plants across m environments, and V_1 is the first row of V.

Figure 3.10 shows the genome profiles obtained using the various methods on four linked traits: days to heading, height, lodging, and yield (full results are available in Appendix A). For bootstrapped methods, 200 iterations were used. From the figure, it is clear the sparsity of the bootstrapped methods seen on the simulated dataset is retained. Stagewise, L^1, and RFE all identified similar sharply defined areas of high variance explained with most markers attributed close to 0% variance explained. In contrast, the BIM profiles were much more noisy with many more markers assigned a high proportion of variance explained, especially for the days to heading phenotype.

Furthermore, the L^1 profiles appear to assign a higher variance explained than the other bootstrap methods for many traits. On all traits excluding yield, the major peaks are assigned a higher variance explained by L^1 than the other bootstrapped methods, and even surpasses MR which is already optimistic. These results suggest the estimation of marker effects can be optimistic for the L^1 method.

Examining the profiles for the days to heading trait shows that there is a second major peak on the second chromosome at approximately 90cM identified by all the bootstrap methods. The same peak was identified to some extent by the BIM method – though many other markers are assigned a higher variance explained such as the markers on the forth chromosome – but was assigned close to 0% variance explained by MR. This indicates the marker is of predictive value, but has no direct correlation with the phenotype, suggesting the whole-genome approach of the bootstrap method provides additional power to detect QTL that are not in direct correlation with the trait of interest.

Finally, the bootstrapped methods identified major peaks coinciding at the same locus for the lodging, yield, and height traits on chromosome 3H. Hayes et al. (1993) suggested the positive allele for the yield QTL on chromosome 3H coincided with low lodging and height QTL alleles from the opposite parent. These previous observations are clearly reinforced by our results and appear to point

to a locus influencing the plant height with independent pleiotropic[6] effects on both lodging and yield as opposed to a causal chain (tall plants → lodging → reduced yield). The plant height also appeared to affect lodging via another QTL on chromosome 2H which coincided for the two traits. In turn, the plant height appeared to be partly associated with heading date because the main QTL on chromosome 2H for these two traits coincide precisely.

On the pubescence trait control, all three bootstrapped methods (stagewise, RFE, and L^1) produced extremely sparse results and correctly identified the single position that characterises the trait (see Figure 3.10). MR identified the same position, but as expected is not as sharply defined due to the high correlation of genetically close markers. Similarly, BIM was unable to locate the QTL as precisely as the bootstrap method, but also vastly underestimated the variance explained.

3.5 Conclusions

The novel methods presented here approach the problem of QTL mapping from an entirely different perspective to traditional techniques. Instead of approaching the task by analysing the entire dataset and performing hypothesis testing, they focus on estimating and attributing the generalisation error of the various markers.

The experiments present strong evidence of the good performance of these algorithms. On synthetic data they were shown to provide better estimates of QTL effects, and identified QTL at least as accurately as the recent BIM method. On natural data the advantages of the bootstrapped methods was further demonstrated. The sparsity of the bootstrapped methods exceeded that obtained by the BIM method for all traits. On the pubescence trait control they clearly outperformed BIM by identifying the single locus with higher definition and more accurate effect estimation. Furthermore, the consistent identification of coinciding QTL across several related traits (days to heading, height, lodging, and yield) with high definition are in agreement with previously hypothesised relationships (Hayes et al., 1993).

[6]The production of two of more effects by a single gene

Algorithm 3.1 The RFE-QTL algorithm

Let:

1. $\Phi\colon \mathbb{R}^m \to \mathbb{R}^{\mathbb{R}^m}$ be the map $\Phi(\boldsymbol{\beta})(\mathbf{x}) := \langle \boldsymbol{\beta}, \mathbf{x} \rangle$;
2. ridge: $2^{\mathbb{R}^m \times \mathbb{R}} \times 2^m \to \mathbb{R}^m$ be the function calculating the ridge regression solution
$$(\mathcal{X}, I) \mapsto (X_I^* X_I + \lambda Id)^{-1} X_I^* \mathbf{y},$$
where $\lambda > 0$, and X_I is the matrix X, but with columns not in I set to zero (i.e., $\mathbf{x}^{(j)} = 0$ if $j \notin I$) ;
3. bootstrap: $2^{\mathbb{R}^m \times \mathbb{R}} \to 2^{\mathbb{R}^m \times \mathbb{R}}$ be a sampling function returning a bootstrap training set;
4. mean: $\mathbb{R}^m \to \mathbb{R}$ be the mean operator;
5. sd: $\mathbb{R}^m \to \mathbb{R}$ be the standard deviation operator;
6. r^2 be defined as in Definition 2.5.

The full RFE-QTL algorithm is as follows:

for $i_{\text{boot}} \in \{1, \ldots, n_{\text{boot}}\}$ **do**
  Obtain a bootstrap split:
  $\mathcal{X}_{\text{train}} \leftarrow \text{bootstrap}(\mathcal{X})$
  $\mathcal{X}_{\text{test}} \leftarrow \mathcal{X} \setminus \mathcal{X}_{\text{train}}$
  The RFE procedure; builds models $\{\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(m)}\}$:
  $J \leftarrow \{1, \ldots, m\}$
  $k \leftarrow 1$
  while $|J| > 0$ **do**
    $\boldsymbol{\beta}^{(k)} \leftarrow \text{ridge}(\mathcal{X}_{\text{train}}, J)$ ▷ Ridge regression solution using features J
    $\hat{\jmath}_k \leftarrow \arg\min_{j \in J} |\beta_{jk}|$ ▷ The "least important" feature
    $J \leftarrow J \setminus \{\hat{\jmath}_k\}$
    $k \leftarrow k + 1$
  end while
  Calculate PAVE
  for $k \in \{1, \ldots, m-1\}$ **do**
    $\text{PAVE}_{\hat{\jmath}_k}^{i_{\text{boot}}} \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(k)})) - r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(k+1)}))$
  end for
  $\text{PAVE}_{j_m}^{i_{\text{boot}}} \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(m)}))$
end for
  Calculate the average PAVE across bootstrap:
for $j \in \{1, \ldots, m\}$ **do**
  $\widehat{\text{PAVE}}_j \leftarrow mean(\text{PAVE}_j)$
end for
return $\widehat{\text{PAVE}}$

Algorithm 3.2 The fast RFE-QTL algorithm

Let the assumptions of Algorithm 3.1 hold. Let $\{o_i\} = \text{order}(\{u_i\})$ be a function that returns a set of indices such that $u_{o_i} \leq u_{o_j}$ iff $i < j$. The fast RFE-QTL algorithm is as follows:

for $i_{\text{boot}} \in \{1, \ldots, n_{\text{boot}}\}$ **do**
  Obtain a bootstrap split:
  $\mathcal{X}_{\text{train}} \leftarrow bootstrap(\mathcal{X})$
  $\mathcal{X}_{\text{test}} \leftarrow \mathcal{X} \setminus \mathcal{X}_{\text{train}}$
  The RFE procedure; builds models $\{\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(m)}\}$:
  $J \leftarrow \{1, \ldots, m\}$
  $k \leftarrow 1$
  while $|J| > 0$ **do**
    $\boldsymbol{\beta}^{(k)} \leftarrow ridge(\mathcal{X}_{\text{train}}, J)$ ▷ Ridge regression solution using features J
    $\{o_i\} \leftarrow order(|\boldsymbol{\beta}^{(k)}|)$ ▷ Order the feature(s) in ascending order
    if $|J| > 100$ **then**
      $C \leftarrow \lceil 0.9|J| \rceil$ ▷ The cutoff point
      $O_k \leftarrow \{o_1, \ldots, o_C\}$
    else
      $O_k \leftarrow \{o_1\}$
    end if
    $J \leftarrow J \setminus O_k$
    $k \leftarrow k + 1$
  end while
  $n_{\text{models}} \leftarrow k - 1$
  Calculate PAVE:
  for $k \in \{1, \ldots, n_{\text{models}} - 1\}$ **do**
    $totvar \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(k)})) - r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(k+1)}))$
    for $j \in O_k$ **do**
      $\text{PAVE}_j^{i_{\text{boot}}} \leftarrow totvar/|O_k|$
    end for
  end for
  $\text{PAVE}_{(O_{n_{\text{models}}})_1}^{i_{\text{boot}}} \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(m)}))$
end for
Calculate average PAVE across bootstrap:
for $j \in \{1, \ldots, m\}$ **do**
  $\widehat{\text{PAVE}}_j \leftarrow mean(\text{PAVE}_j)$
end for
return $\widehat{\text{PAVE}}$

Algorithm 3.3 The full L^1 algorithm

Let:

1. Φ, bootstrap, mean, sd, and r^2 be defined as in Algorithm 3.1;
2. $L1\colon 2^{\mathbb{R}^m \times \mathbb{R}} \times \mathbb{R}^+ \to \mathbb{R}^m$ be the function returning the L^1 empirical risk minimiser.

The L^1 algorithm is as follows:

for $i_{\text{boot}} \in \{1, \ldots, n_{\text{boot}}\}$ **do**
  Obtain a bootstrap split:
  $\mathcal{X}_{\text{train}} \leftarrow \text{bootstrap}(\mathcal{X})$
  $\mathcal{X}_{\text{test}} \leftarrow \mathcal{X} \setminus \mathcal{X}_{\text{train}}$
  Hyperparameter tuning:
  $\mathcal{X}'_{\text{train}} \leftarrow \text{bootstrap}(\mathcal{X}_{\text{train}})$
  $\mathcal{X}'_{\text{test}} \leftarrow \mathcal{X}_{\text{train}} \setminus \mathcal{X}'_{\text{train}}$
  for each λ **do**
    $r_\lambda \leftarrow r^2(\mathcal{X}'_{\text{test}}, \Phi(L1(\mathcal{X}'_{\text{train}}, \lambda)))$
  end for
  $\hat{\lambda} \leftarrow \arg\max_\lambda r_\lambda$
  Build final model:
  $\beta^{(i_{\text{boot}})} \leftarrow L1(\mathcal{X}_{\text{train}}, \hat{\lambda})$
  Calculate PAVE:
  for $j \in \{1, \ldots, m\}$ **do**
    $\text{PAVE}_j^{(i_{\text{boot}})} \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(m)})) \frac{\left|\beta_j^{(i_{\text{boot}})}\right|}{\left|\left|\boldsymbol{\beta}^{(i_{\text{boot}})}\right|\right|_2}$
  end for
end for
  Calculate average PAVE across bootstrap:
$\widehat{\text{PAVE}}_j \leftarrow \text{mean}(\text{PAVE}_j)$
return $\widehat{\text{PAVE}}$

Algorithm 3.4 The L^1-BIC algorithm

Let:

1. Φ, bootstrap, mean, sd, and r^2 be defined as in Algorithm 3.1;
2. $L1\colon 2^{\mathbb{R}^m \times \mathbb{R}} \times \mathbb{R}^+ \to \mathbb{R}^m$ be the function returning the L^1 empirical risk minimiser;
3. BIC: $\mathbb{R}^m \to \mathbb{R}$ be the function calculating the BIC given a model on the training data $\mathcal{X}_{\text{train}}$ (see Definition 2.32).

The L^1-BIC algorithm is as follows:

for $i_{\text{boot}} \in \{1, \ldots, n_{\text{boot}}\}$ **do**
  Obtain a bootstrap split:
  $\mathcal{X}_{\text{train}} \leftarrow \text{bootstrap}(\mathcal{X})$
  $\mathcal{X}_{\text{test}} \leftarrow \mathcal{X} \setminus \mathcal{X}_{\text{train}}$
  Hyperparameter tuning using the BIC:
  for each λ **do**
    $\boldsymbol{\beta}^{\lambda} \leftarrow L1(\mathcal{X}_{\text{train}}, \lambda)$
    $r_{\lambda} \leftarrow \text{BIC}(\boldsymbol{\beta}^{\lambda})$
  end for
  $\hat{\lambda} \leftarrow \arg\max_{\lambda} r_{\lambda}$
  $\boldsymbol{\beta}^{(i_{\text{boot}})} \leftarrow \boldsymbol{\beta}^{\lambda}$
  Calculate PAVE:
  for $j \in \{1, \ldots, m\}$ **do**
    $\text{PAVE}_j^{(i_{\text{boot}})} \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(m)})) \frac{\left|\beta_j^{(i_{\text{boot}})}\right|}{\left|\left|\boldsymbol{\beta}^{(i_{\text{boot}})}\right|\right|_2}$
  end for
end for
Calculate average PAVE across bootstrap:
$\widehat{\text{PAVE}}_j \leftarrow \text{mean}(\text{PAVE}_j)$
return $\widehat{\text{PAVE}}$

Algorithm 3.5 The Stagewise algorithm

Let:

1. Φ, bootstrap, mean, sd, and r^2 be defined as in Algorithm 3.1;
2. BIC: $\mathbb{R}^m \to \mathbb{R}$ be the function calculating the BIC given a model on the training data $\mathcal{X}_{\text{train}}$.

The stagewise algorithm is defined as follows:

for $i_{\text{boot}} \in \{1, \ldots, n_{\text{boot}}\}$ **do**
  Obtain a bootstrap split:
  $\mathcal{X}_{\text{train}} \leftarrow \text{bootstrap}(\mathcal{X})$
  $\mathcal{X}_{\text{test}} \leftarrow \mathcal{X} \setminus \mathcal{X}_{\text{train}}$
  Stagewise regression:
  $\boldsymbol{\beta} \leftarrow 0$
  repeat
  Calculate residual:
    for $i \in \{1, \ldots, n\}$ **do**
      $res_i \leftarrow y_i - \langle \mathbf{x}_i, \boldsymbol{\beta} \rangle$
    end for
    $c_i \leftarrow \text{cor}(\mathbf{x}^{(i)}, \mathbf{res})$
    $\hat{\imath} \leftarrow \arg\max_i |c_i|$
    $\beta_{\hat{\imath}} \leftarrow \beta_{\hat{\imath}} + \Delta \, \text{sign}(c_{\hat{\imath}})$
  until Termination conditions are met
  Calculate PAVE:
  for $j \in \{1, \ldots, m\}$ **do**
    $\text{PAVE}_j^{(i_{\text{boot}})} \leftarrow r^2(\mathcal{X}_{\text{test}}, \Phi(\boldsymbol{\beta}^{(m)})) \frac{\left|\beta_j^{(i_{\text{boot}})}\right|}{\left|\left|\boldsymbol{\beta}^{(i_{\text{boot}})}\right|\right|_2}$
  end for
end for
Calculate average PAVE across bootstrap:
$\widehat{\text{PAVE}}_j \leftarrow \text{mean}(\text{PAVE}_j)$
return $\widehat{\text{PAVE}}$

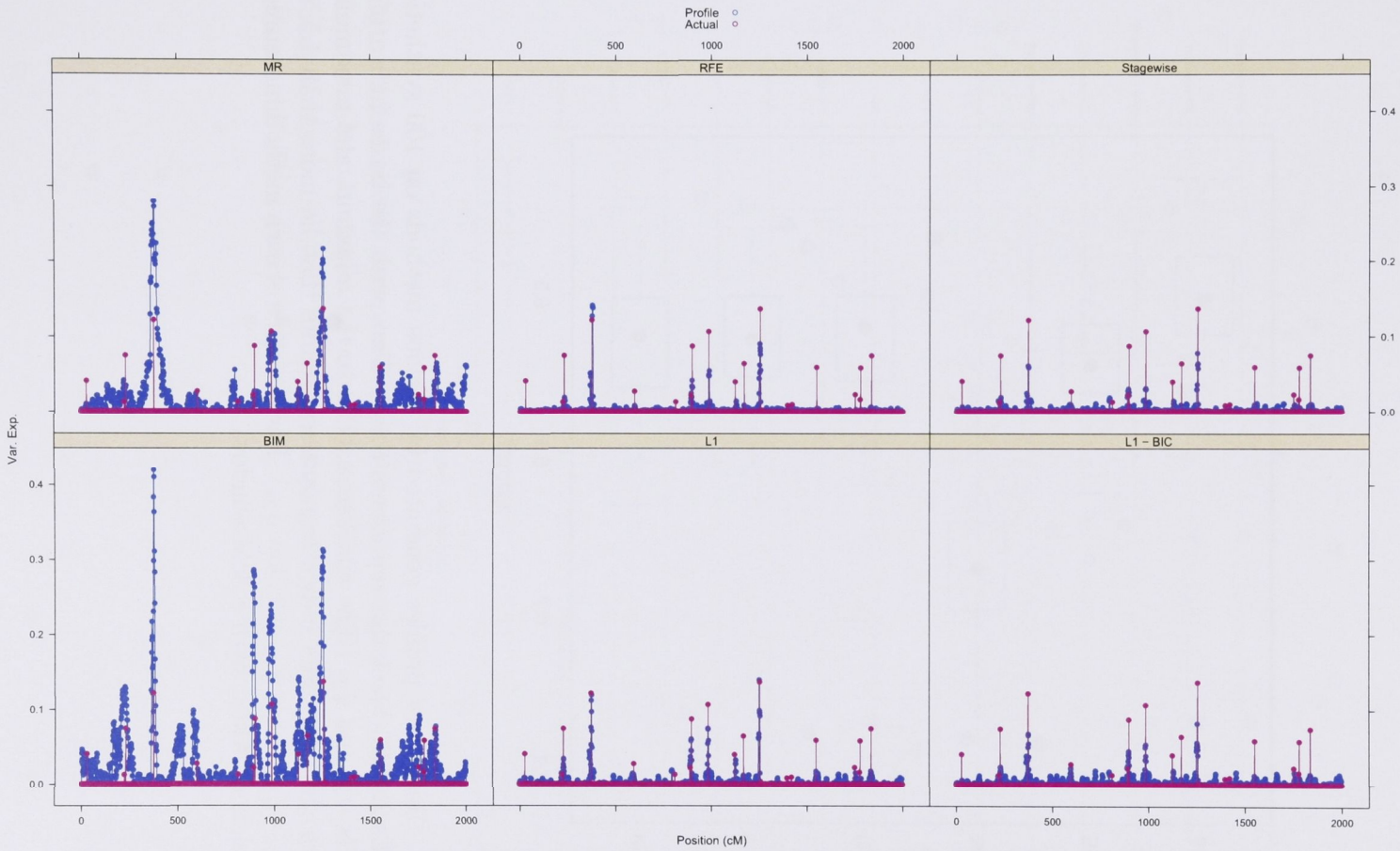


Figure 3.7: Synthetic dataset analysis. Purple plot lines indicate true QTL locations and effects. Individual boxes show profiles generated by the different methods.

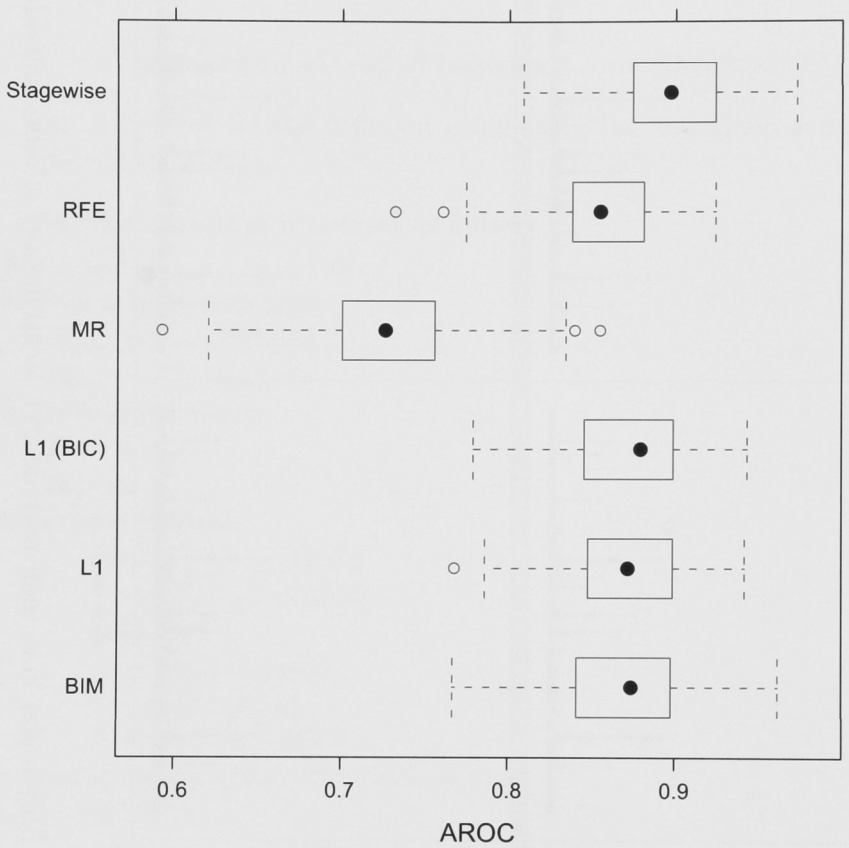


Figure 3.8: Results of profiles generated by various methods on 100 synthetic datasets. Each solid black dot represents the median, with the boxes indicating the quartiles $q_{.75}$ and $q_{.25}$. The whiskers extend to the minimum and maximum values, with the open dots indicating outliers (points that lie more than $1.5 \times IQR := q_{.75} - q_{.25}$ above $q_{.75}$ or below $q_{.25}$). Performance of each method measured using a "relaxed" AROC with a 3cM window.

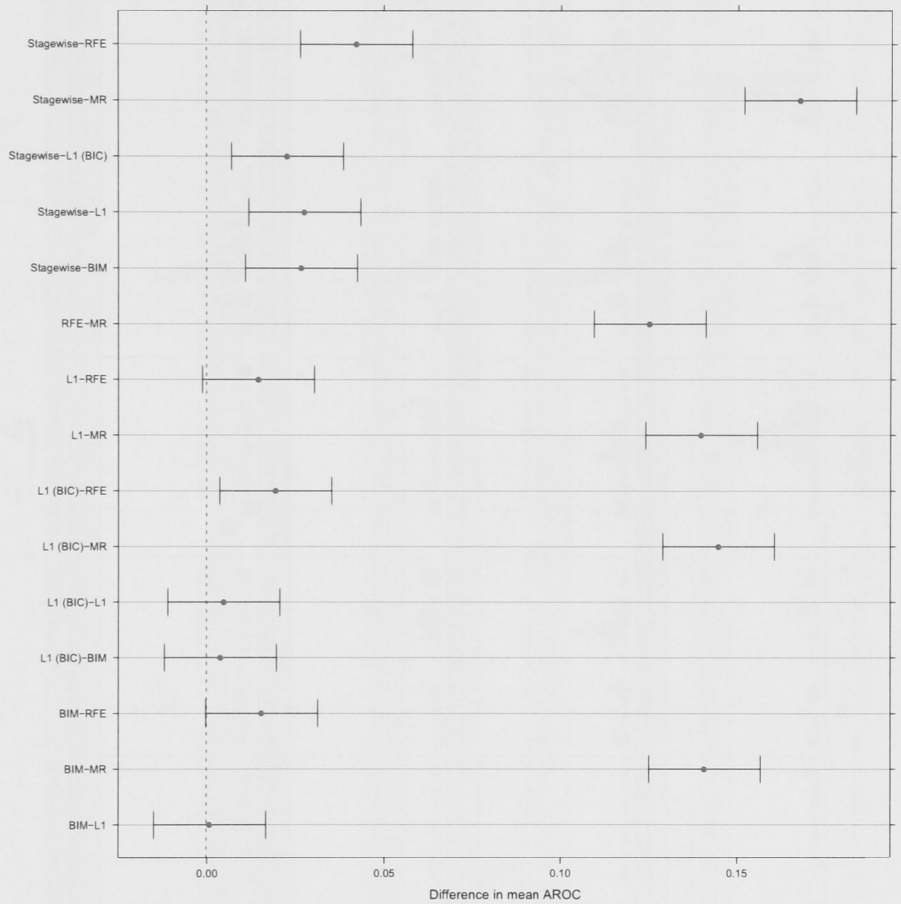


Figure 3.9: Pairwise comparisons of difference between means using the Tukey-Kramer method. Intervals are 95% confidence intervals calculated using the Tukey-Kramer method. Intervals not containing 0 indicate a statistically significant difference at the 95% level.

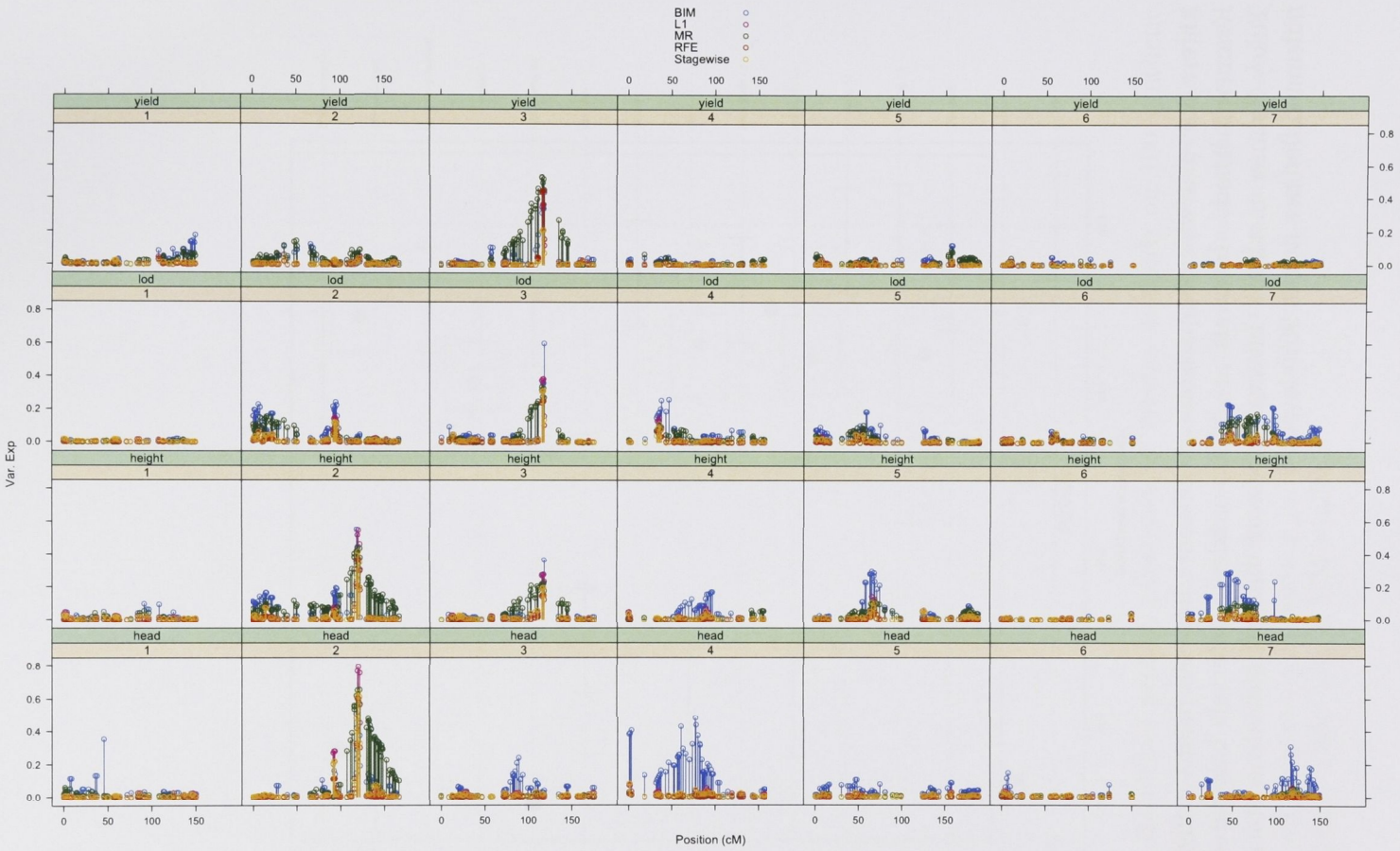


Figure 3.10: ϵ-0 bootstrap results with 200 iterations for four phenotypes (days to heading, height, yield, and lodging) on barley data. Genome profiles generated using several different methods: Stagewise, RFE, L^1, and BIM. The y-axis is variance explained and the x-axis is chromosome position in cM.

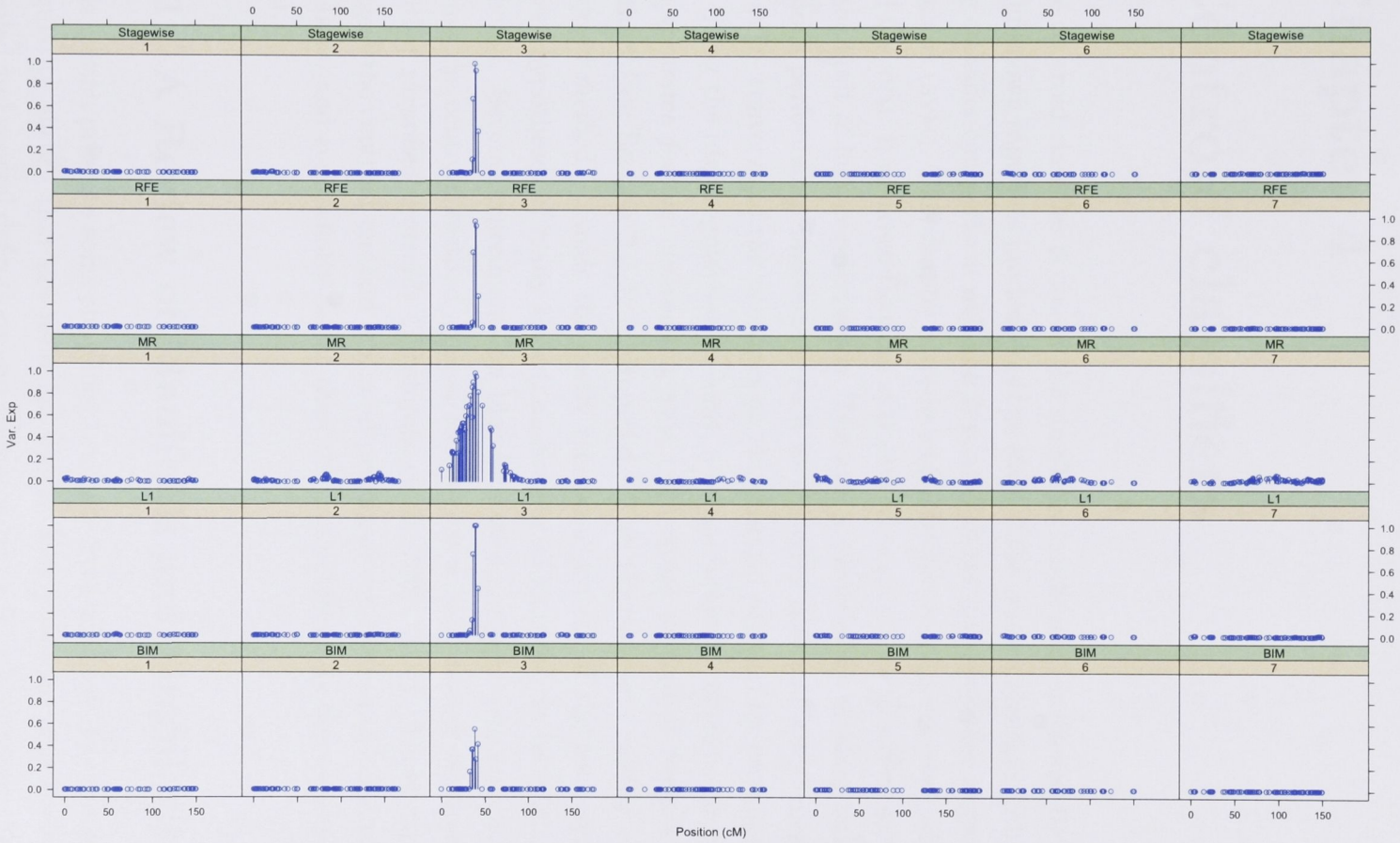


Figure 3.11: Results for the pubescence phenotype on DArT natural data. Details are as Figure 3.10.

Chapter 4

Centroid classifiers

The centroid classifier is one of the simplest classification methods; the class for an unknown sample is predicted as the class of the nearest centroid. Surprisingly, this classifier can perform well and is related to the more complex support vector machine (SVM). This chapter introduces formalities linking the centroid classifier and the SVM: it is shown that the SVM converges pointwise to a centroid classifier in the limit of high-regularisation. The link is extended by showing that discontinuous performance measures – such as the error rate and AROC measured on an SVM– converge to the measures on the centroid solution for certain maps.

Using the high-regularisation limit, a computationally efficient alternative to the *recursive feature elimination* SVM (RFE–SVM) embedded feature selection method (see Section 2.5.3) is proposed. The alternative – hereby called the *centroid method* – avoids the costly recursion process of the RFE–SVM, which allows hypotheses of many sizes (i.e., number of features) to be efficiently constructed. Several experiments on bioinformatics datasets were conducted comparing the centroid method against the RFE–SVM and another centroid method entitled *shrunken centroids* (Tibshirani et al., 2002, 2003). These experiments suggest the centroid method is a good performer on small-sample bioinformatics datasets, and significantly faster than the computationally more complex RFE–SVM.

4.1 A Review of Manifolds and Singularities

This section presents some simplified concepts from topology, transversality, and singularity theory that are used for stating later theorems and proofs. More detailed and general definitions and theorems can be found elsewhere (Demazure,

2000; Golubitsky and Guillemin, 1974; Hocking and Young, 1988; Willard, 2004).

4.1.1 Topological Spaces

The presentation here follows Willard (2004).

Definition 4.1 (Topological space). *A topological space is a set X together with a collection of subsets T such that*

- *the empty set and the whole space X are contained in T;*
- *the union $\bigcup_{i \in I} U_i$ of any collection of subsets from T is contained in T;*
- *the intersection $\bigcap_{i \in I} U_i$ of any finite collection of subsets $\{U_i\}_{i \in I} \subset T$ is contained in T.*

The set T is called a topology on X, and sets $U \in T$ are called the open sets. A subset $U \subset X$ is called closed iff $X \setminus U \in T$.

Given this definition, a basic topology over $\mathbb{R}^n$ is the smallest collection of sets satisfying the above conditions and containing all open balls

$$B_r := \{\mathbf{x} : ||\mathbf{x} - \mathbf{x}_0||_2 < r\}$$

for $r > 0$ and $\mathbf{x}_0 \in \mathbb{R}^n$. A topological space (X, T) is abbreviated as X if there is no confusion over the topology T. Given two topologies T_1, T_2 over the same set X, T_1 is called *weaker* than T_2 and T_2 is called *stronger* than T_1 if $T_1 \subset T_2$.

The *base* B of a topology (X, T) is a collection of open sets such that for every $U \in T$, there exists a collection $\{V_i\} \subset B$ such that $U = \bigcup_i V_i$. If there exists a countable base B, then (X, T) is said to be a *second countable space*.

Definition 4.2. *Given a topological space X, a neighbourhood of a point $x \in X$ is a set V such that*

$$x \in U \subseteq V$$

for an open set U.

Definition 4.3 (Hausdorff space). *A Hausdorff space (or T_2 space) is a topological space X such that for any $x, x' \in X$, there exists a neighbourhood U of x and a neighbourhood U' of x' such that $U \cap U' = \emptyset$.*

Non-Hausdorff spaces are not studied in this thesis.

Definition 4.4 (Images). *Let $f: U \to V$. The set*

$$f[U] := \{v \in V | \exists u \in U \quad \textit{such that} \quad f(u) = v\}$$

is called the image *of the map f, and*

$$f^{-1}[V] := \{u \in U | \exists v \in V \quad \textit{such that} \quad f(u) = v\}$$

is called the inverse image.

Note that the inverse image $f^{-1}[V]$ is not the same as the image of the inverse function unless f is a bijection. The inverse image of a singleton (e.g., $f^{-1}[0] := f^{-1}[\{0\}]$) is called a *fibre*.

Definition 4.5 (Homeomorphisms). *Let X and Y be topological spaces and $f: X \to Y$. The map f is a homeomorphism iff it has the following properties:*

1. *f is a bijection, i.e., for every $y \in Y$ there is exactly one $x \in X$ such that $f(x) = y$;*
2. *f is continuous;*
3. *the inverse function f^{-1} is continuous.*

Homeomorphisms are useful as they map open sets to open sets, and closed sets to closed sets. Furthermore, homeomorphic spaces share many properties such as compactness, which is defined in a later subsection.

4.1.2 Differentiable Mappings

Let U be a subset of $\mathbb{R}^n$ and denote its interior by $\text{Int}(U) := \bigcup\{V \in T | V \subset U\}$ and its closure by $\overline{U} := X - \text{Int}(X - U)$. Furthermore, let $f: U \to \mathbb{R}$, and $\mathbf{u} \in U$. The notation $(\frac{\partial}{\partial u_i})$ is used to refer to the partial derivative of the i^{th} component of $\mathbf{u}$. The higher order mixed partial derivatives are denoted as

$$\partial_{\boldsymbol{\alpha}} f := \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial u^{\boldsymbol{\alpha}}} f = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial u_1^{\alpha_1} \partial u_2^{\alpha_2} \ldots \partial u_n^{\alpha_n}} f,$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ as an n-tuple of non-negative integers and $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 + \ldots \alpha_n$.

Two classes of functions studied herein are *smooth functions* and *real analytic functions*.

Definition 4.6 (Smooth and analytic functions). *Let k be a positive integer. A function f is said to be of class C^k if $(\frac{\partial^{|\boldsymbol{\alpha}|} f}{\partial u^{\boldsymbol{\alpha}}})(u)$ exists and is continuous for every non-negative tuple $\boldsymbol{\alpha}$ such that $|\boldsymbol{\alpha}| := ||\boldsymbol{\alpha}||_1 \leq k$ and every u. A C^∞ function is called smooth. A C^∞ function f is called* real analytic *if for all u_0, the Taylor series expansion of f converges to f for u sufficiently close to u_0, i.e.,*

$$f(u) = f(u_0) + \sum_{k=1,|\boldsymbol{\alpha}|\leq k}^{\infty} \frac{1}{\boldsymbol{\alpha}!} \partial_{\boldsymbol{\alpha}} f(u_0)(u - u_0)^{\boldsymbol{\alpha}}$$

where $u = (u_1, \ldots, u_n)$, the operator $u^{\boldsymbol{\alpha}} := u_1^{\alpha_1} u_2^{\alpha_2} \cdots u_n^{\alpha_n}$, and $\boldsymbol{\alpha}! := \alpha_1! \alpha_2! \cdots \alpha_n!$. The space of real analytic functions is denoted C^ω.

Note that there are many smooth but non-analytic functions. An example is the bump function defined later in Lemma 4.18; this function is smooth, but not analytic at the boundaries $\pm r$.

The next theorem is a fundamental result in mathematical analysis. It states that given a C^1 map, there exists an inverse map if the Jacobian has full rank[1]. This theorem is used directly in convergence proofs in Chapter 4.

Theorem 4.7 (Inverse Function Theorem). *Let $U \subset \mathbb{R}^n$ be open, $u \in U$, and $\phi \colon U \to R^m$ be a C^k mapping. If the Jacobian*

$$J_\phi(u) := [\partial_{u_i} \phi_j(u)]_{1 \leq i \leq n; 1 \leq j \leq m}$$

has full rank, then there exists an open set $V \subset \phi[U]$ and a C^k map $\psi \colon V \to U$ such that

$$\phi \circ \psi(v) = v \ \forall v \in V$$

and

$$\psi \circ \phi(u) = u \ \forall u \in \psi[V]$$

Proof. See Demazure (2000). □

4.1.3 Manifolds

Definition 4.8 (Manifold). *A manifold is a second countable Hausdorff space X such that for every $x \in X$, there exists an open set $U_x \subset X$ containing x and a*

[1] Recall the matrix rank is the number of linearly independent rows or columns, and a $n \times m$ matrix has full rank if the rank is $min(n, m)$.

homeomorphism $\phi_x : U_x \to \mathbb{R}^n$ known as a chart (Willard, 2004). A C^k manifold (for $k \in \{1, 2, \ldots, \infty\}$) has the additional restriction that given $x, y \in X$, there exists charts ϕ_x and ϕ_y such that if $\phi_x^{-1}[V_x] \cap \phi_y^{-1}[V_y] \neq \emptyset$, then $\phi_y \circ \phi_x^{-1}$ is a C^k map.

Manifolds may be generalised to Hausdorff spaces that are not second countable, but this generalisation is unnecessary for the spaces studied herein. Intuitively, manifolds are spaces where a small local area "looks" like normal Euclidean space. The advantage of this is that standard Euclidean geometry can be applied to small local areas of manifolds.

Definition 4.9 (Cover). *A collection of sets $\{U_i\}_{i \in I}$ is a* cover *of a space X iff $U_i \subset X$ for all i, and $\bigcup_{i \in I} U_i = X$. If all the sets U_i are open, the cover is called an* open cover.

Definition 4.10 (Submanifold). *Let M be a manifold of dimension m. Then, N is a submanifold of M with dimension $n \in [1, m]$ iff there exists a chart $\phi : U \subset M \to \mathbb{R}^m$ such that $\phi[U \cap N] \subset \{(v_i) \in \mathbb{R}^m | v_i = 0 \text{ for } n < i \leq m\}$. N is a C^k-submanifold if ϕ is a C^k diffeomorphism. The difference between dimensions is called the codimension of N and is denoted* $\operatorname{codim}(N) := m - n$.

Note that submanifold N of $M \subset \mathbb{R}^m$ has Lebesgue measure 0 in $\mathbb{R}^m$ if $\operatorname{codim}(N) > 0$[2].

4.1.4 Compactness

The concept of paracompactness is now introduced, and all manifolds are shown to be paracompact. The paracompactness property guarantees the existence of *partitions of unity*, which are defined in the following section.

Definition 4.11 (Refinement). *Let $U = \{U_i\}_{i \in I}$ and $V = \{V_j\}_{j \in J}$ be two covers of a space X. Then, it is said U refines V iff for every V_j, there exists a U_i such that $U_i \subset V_j$.*

Definition 4.12 (Locally compact (Willard, 2004)). *A space X is locally compact iff every point $x \in X$ has a compact neighbourhood.*

Corollary 4.13. *Manifolds are locally compact.*

[2]This can be shown as the image of every chart $\phi_n[U_n \cap N]$ is negligible in R^m

Proof. Let X be a manifold. By definition, every point $x \in X$ has a neighbourhood homeomorphic by the chart ϕ_x to $\mathbb{R}^n$. As $\mathbb{R}^n$ is locally compact and ϕ_x is a homeomorphism, X is locally compact. □

Definition 4.14 (Locally finite (Willard, 2004)). *A collection of subsets $\{U_i\}_{i \in I}$ of X is locally finite iff for all $x \in X$ there exists a neighbourhood V_x of x such that the set*

$$\{i \in I | U_i \cap V_x \neq \emptyset\}$$

is finite.

Definition 4.15 (Paracompactness (Willard, 2004)). *A Hausdorff space X is paracompact iff each open cover of X has an open locally finite refinement.*

It follows from the definitions of compactness and paracompactness that every compact set is paracompact.

Theorem 4.16. *If X is a second countable locally compact space then it is paracompact.*

Proof. See (Willard, 2004). □

A consequence of this theorem is that all manifolds are paracompact.

Partitions of unity

Definition 4.17 (Partition of unity (Golubitsky and Guillemin, 1974)). *Let X be a manifold. A partition of unity subordinate to the open cover $\{U_i\}_{i \in I}$ is a set of continuous functions $\{\rho_j\}_{j \in J}$ defined on X such that for any $x \in X$:*

1. *There exists a neighbourhood of x such that a finite number of functions ρ_j are non-zero;*
2. *The sum $\sum_{j \in J} \rho_j(x) = 1$.*
3. *The collection $V_j := \rho_j^{-1}[(0, \infty)] = \{x \in X | \rho_j(x) > 0\}$ is a refinement of $\{U_i\}$.*

Partitions of unity are used for a proof in Chapter 4, and they exist for any manifold, as will be proved shortly.

Lemma 4.18. *For any open ball $B_r \subset \mathbb{R}^n$ centred at $\mathbf{x}_0$ with a radius r, there exists a smooth bump function $\psi \colon \mathbb{R}^n \to \mathbb{R}$ that is positive everywhere on B_r and zero off B_r.*

Proof. Define $\phi\colon \mathbb{R} \to \mathbb{R}$ by

$$x \mapsto \begin{cases} e^{\frac{-1}{1-x^2}} & \text{for } x^2 < 1 \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see this function is smooth, positive on the interval $(-1, 1)$, and zero otherwise. Defining the function

$$\psi(\mathbf{x}) := \phi(\frac{||\mathbf{x} - \mathbf{x}_0||_2^2}{r^2})$$

completes the proof. □

Theorem 4.19. *Let X be a manifold and $\{U_i\}_{i\in I}$ be an open covering. There exists a partition of unity $\{\rho_j\}_{j\in J}$ subordinate to the open cover $\{U_i\}_{i\in I}$.*

Proof. Let $\{V_j\}_{j\in J}$ be a locally finite refinement of $\{U_i\}_{i\in I}$ guaranteed to exist as X is paracompact. Define $g_j\colon X \to R$ by

$$v \mapsto \begin{cases} \gamma(v) & \text{if } v \in V_j \\ 0 & \text{otherwise} \end{cases}$$

where γ is a smooth function positive on V_j and zero off V_j using Lemma 4.18. Let $h := \sum_{j\in J} g_j$. The function h is well defined, C^∞, and always positive as $\{V_j\}_{j\in J}$ is a locally finite covering of X. Defining the function

$$\rho_j := \frac{g_j}{h}$$

yields the smooth partition of unity. □

4.1.5 Jets and Transversality

Jets and transversality are two concepts used again for convergence proofs in Chapter 4. The treatment given here is necessarily brief; more information can be found from other sources (Demazure, 2000; Golubitsky and Guillemin, 1974).

Definition 4.20 (Jets and jet spaces)**.** *Let $P^k(\mathbb{R}^n; \mathbb{R}^m)$ denote the space of polynomial functions $f\colon \mathbb{R}^n \to \mathbb{R}^m$ with degree k. Furthermore, let $U \subset \mathbb{R}^n$ and $f\colon U \to \mathbb{R}^m$ be of class C^{k+1}. Define the space of jets with order k as*

$$J^k(U; \mathbb{R}^m) = U \times P^k(\mathbb{R}^n; \mathbb{R}^m).$$

The k-jet of f about u_0, denoted $(j^k_{u_0} f)$, is the truncated Taylor expansion of f of degree k about u_0:

$$(j^k_{u_0} f)(z) := \sum_{|\boldsymbol{\alpha}| \leq k} \frac{1}{\boldsymbol{\alpha}!} \partial_{\boldsymbol{\alpha}} f(u_0) z^{\boldsymbol{\alpha}}$$

The maps

$$j^k_* f \colon U \to P^k(U; \mathbb{R}^m), \ (j^k_* f)(x) = j^k_x f$$
$$j^k f \colon U \to J^k(U; \mathbb{R}^m), \ (j^k f)(x) = (x, j^k_x f)$$

are the k-jets of f.

In particular, $j^0 f(a) = (a, f(a))$ and $j^1 f(a) = (a, f(a), df(a))$ where $df(a) := (\partial_{a_1} f(a), \partial_{a_2} f(a), \ldots, \partial_{a_n} f(a))$.

Definition 4.21 (Tangent space)**.** *Let X be a C^k manifold with $k \geq 1$ and $x \in X$. Let $\phi \colon U \to \mathbb{R}^n$ be a chart where U is an open subset of X containing x. Consider curves (C^1 maps) of the form $\gamma \colon (-1, 1) \to X$ where $\gamma(0) = x$. Two curves γ and γ' are considered tangent if $(d(\phi \circ \gamma))(0) = (d(\phi \circ \gamma'))(0)$. The tangent space of X about x, denoted $T_x X$, is the equivalence class defined by this equivalence relation. It has the natural structure of a vector space.*

Definition 4.22 (Transversality)**.** *Let X be a manifold with two submanifolds N and M. The submanifolds N and M intersect transversally iff for all $x \in N \cap M$,* $\operatorname{span}(T_x N \cup T_x M) = T_x X$[3]*. Equivalently, the submanifolds intersect transversally iff* $\operatorname{codim}(N \cap M) = \operatorname{codim}(N) + \operatorname{codim}(M)$.

Definition 4.23 (Meagre and residual sets)**.** *A subset $U \subset X$ of a topological space X is called* meagre *iff there exists a countable cover $\{V_i\}_{i=1}^{\infty}$ such that* $\operatorname{Int}(\overline{V}) = \emptyset$ *for all i. The complement of a meagre set is called a* residual *set.*

Theorem 4.24 (Thom's Transversality Theorem)**.** *Let $U \subset \mathbb{R}^n$ and W be a submanifold of $J^k(U; \mathbb{R}^m)$. The set of polynomials $p \in P^k(\mathbb{R}^n; \mathbb{R}^m)$ such that $j^k(f + p)$ is transverse to W is a residual subset of $P^k(\mathbb{R}^n; \mathbb{R}^m)$ with negligible complement. If W is closed, the set of p such that $j^k(f + p)$ is transverse to W on a compact subset $K \subset U$ is an open dense subset of $P^k(\mathbb{R}^n; \mathbb{R}^m)$ with negligible complement.*

Proof. See Demazure (2000). □

[3]Recall the span of S is the collection of all finite linear combinations of elements in S.

4.2 Empirical Risk Minimisation

This chapter studies hypotheses obtained through minimising a generalised version of the empirical regularised risk introduced in Section 2.4, in particular the SVM. The generalisation allows *class specific regularisation*, through which a *single class* SVM can be obtained with an appropriate choice of loss function and hyperparameters.

Let the given data be $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{X} \times \{1, -1\}$ and $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a reproducing kernel with Hilbert space $\mathcal{H}$. Denote the hypothesis as

$$f(\mathbf{x}) := f_H(\mathbf{x}) + \mu_0 = \sum_{i=1}^{n} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + \mu_0, \tag{4.1}$$

where f_H is the *homogeneous* portion and $\mu_0 \in \mathbb{R}$ is the bias. f_H and μ_0 are defined as the minimum of the class dependent regularised risk functional $(f_H, \mu_0) = \arg\min_{f_H, \mu_0} R[f_H, \mu_0]$, where

$$R[f_H, \mu_0] := \sum_{i=1}^{n} [C_{y_i} L(y_i, f_H(\mathbf{x}_i) + \mu_0)] + \Omega(f_H, \mu_0). \tag{4.2}$$

Here, C_y has replaced the regularisation constant λ previously used (see Definition 2.19). When $C = C_- = C_+$, then the equivalence $\lambda = \frac{1}{C}$ holds, but this does not apply when $C_- \neq C_+$.

The class dependent regularisation constants can always be written in the form

$$C_y = \frac{1 + yB}{2|I_y|} C, \tag{4.3}$$

where $I_+ = I_{+1} := \{i | y_i = 1\}$, $I_- = I_{-1} := \{i | y_i = -1\}$, and

$$C := C_- |I_-| + C_+ |I_+| > 0.$$

The choice of $B = 0$ balances the regularisation constant according to the empirical class proportions in the training set, and the choice of

$$B = \frac{|I_+| - |I_-|}{|I_+| + |I_-|}$$

is equivalent to a single regularisation constant (i.e., $C_+ = C_-$). Note that a

single class SVM is possible by choosing $B = \pm 1$.

The SVM solution is given by minimising the regularised risk with the soft-margin loss function

$$L(y, \xi) := \max(0, 1 - y\xi)^p$$

and L^2 regulariser $\Omega(f) := ||f_H||^2$. The *hinge-loss* SVM (denoted SVM1) is given when $p = 1$, and the *quadratic-loss* SVM (denoted SVM2) with $p = 2$. The *homogeneous solution* is found by minimising $R[f_H, 0]$, and similarly the full solution found by minimising $R[f_H, \mu_0]$.

Definition 4.25 (Centroid projection)**.** *The* centroid projection *is*

$$g_B(\mathbf{x}) := \frac{1+B}{2|I_+|} \sum_{i \in I_+} k(\mathbf{x}, \mathbf{x}_i) - \frac{1-B}{2|I_-|} \sum_{i \in I_-} k(\mathbf{x}, \mathbf{x}_i) \tag{4.4}$$

for any $\mathbf{x} \in \mathbb{R}^m$ *and* B *such that* $-1 \leq B \leq 1$*. In the linear case this is equivalent to a linear prediction function* $g_B(\mathbf{x}) = \langle \mathbf{x}, \boldsymbol{\beta}_B^{cent} \rangle$ *with*

$$\boldsymbol{\beta}_B^{cent} := \frac{1+B}{2} \bar{\mathbf{x}}_+ - \frac{1-B}{2} \bar{\mathbf{x}}_-,$$

where $\bar{\mathbf{x}}_+$ *and* $\bar{\mathbf{x}}_-$ *are the centroids of the positive and negative class.*

This is the projection on to the vector connecting the (weighted) arithmetic means of samples from both class labels, or when $B = \pm 1$ the vector connecting the origin and a single class. Equation 4.4 is equivalent to the difference between Parzen window density estimates of the class probabilities with an appropriate choice of k. There is also a link between the centroid projection and the empirical *maximum mean discrepancy* (MMD) introduced in Section 5.1; it is the function which maximally discriminates between the probability distributions generating the two classes.

4.3 Pointwise Convergence

This section presents formal results proving the pointwise convergence of the SVM to the centroid classifier. The proof covers a variety of combinations between loss functions and regularised risk functionals. These combinations are explicitly stated in the following definition.

Definition 4.26. *The three variants of class dependent regularised risk are:*

$\mathcal{R}_0$: *Assume* $|B| < 1$ *with loss function*

$$L(y,\xi) := (1-y\xi)^2 \text{ or}$$
$$L(y,\xi) := \max(0, 1-y\xi)^2,$$

with the regularised risk functional

$$R[f_H, \mu_0] := C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} L(y, f_H(\mathbf{x}) + \mu_0) + ||f_H||^2_{\mathcal{H}}. \tag{4.5}$$

$\mathcal{R}_{\text{hom}}$: *Assume* $|B| \leq 1$ *with loss function*

$$L(y,\xi) := (1-y\xi)^2,$$
$$L(y,\xi) := \max(0, 1-y\xi), \text{ or}$$
$$L(y,\xi) := \max(0, 1-y\xi)^2,$$

with the regularised risk functional

$$R[f_H] := C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} L(y, f_H(\mathbf{x})) + ||f_H||^2_{\mathcal{H}}. \tag{4.6}$$

$\mathcal{R}_{bound}$: *Assume* $|B| \leq 1$ *with loss function as in the* $\mathcal{R}_{\text{hom}}$ *case with the regularised risk functional*

$$R[f_H, \mu_0] := C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} L(y, f(\mathbf{x})) + ||f_H||^2_{\mathcal{H}} + \mu_0^2. \tag{4.7}$$

Note the differences between the three regularised risk solutions. The first, $\mathcal{R}_0$, is the typical ridge regression or SVM solution, but excludes the linear hinge-loss $L(y,\xi) = \max(0, 1-y\xi)$ – the most popular form of SVM– and the single class case when $B = \pm 1$. This exclusion is necessary as the convergence theorems do not hold when using the linear hinge-loss (see Example 4.27). The second set of assumptions, $\mathcal{R}_{\text{hom}}$, allows all values of $-1 \leq B \leq 1$, including the single class case $B = \pm 1$, and all three forms of loss (least squares, linear and quadratic hinge-loss), but applies only to *homogeneous classifiers* ($f_{H,C}$). The final set of assumptions, $\mathcal{R}_{\text{bound}}$, also allows all three forms of loss and the whole range of $|B| \leq 1$, but there is the additional regularisation term over the bias μ_0^2. This is named the *bounded* case. Finally, though the loss $L(y,\xi) = (1-y\xi)^2$ is

counter-intuitive as it penalises points that are "very correct," it is nevertheless particularly important as the other loss functions reduce to this case in the limit of high regularisation.

Example 4.27 (The non-convergence of the hinge-loss SVM)**.** *Consider the 3-point dataset*

$$\mathcal{X} := \{(\mathbf{x}_i, y_i)\}_{i=1}^{3} \subset \mathbb{R}^n \times \{1, -1\}$$

and a linear predictor

$$f(\mathbf{x}; \boldsymbol{\beta}) := \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0$$

with centroid solution $\boldsymbol{\beta}_B^{cent}$ *and hinge-loss* SVM *solution* $\boldsymbol{\beta}^{\text{SVM}1}$*. For simplicity, let* $y_1 = y_2 = -y_3 = 1$ *and* $\mathbf{x}_3 = 0$*. In this case the centroid solution* $\boldsymbol{\beta}_B^{cent}$ *is aligned with the average of the first two data points, i.e.,*

$$\boldsymbol{\beta}_B^{cent} \sim \frac{\mathbf{x}_1 + \mathbf{x}_2}{2}.$$

Consider the case where $B = 1/3$*, which results in equal regularisation constants* $C_{y_i} = \frac{C}{3} =: C'$ *for all* i*. Recall from Example 2.23 that the linear hinge loss* SVM *solution has the expansion* $\boldsymbol{\beta}^{\text{SVM}1} = \sum_i y_i \alpha_i \mathbf{x}_i$*, where coefficients* α_i *must satisfy the conditions*

$$\sum_i \alpha_i y_i = 0$$

and

$$\alpha_i = \begin{cases} C' & \text{if } y_i f(\mathbf{x}_i; \boldsymbol{\beta}^{\text{SVM}1}) < 1 \\ \in (0, C'] & \text{if } y_i f(\mathbf{x}_i; \boldsymbol{\beta}^{\text{SVM}1}) = 1 \\ 0 & \text{otherwise} \end{cases}$$

It follows that there are only three possible directions for the vector $\boldsymbol{\beta}^{\text{SVM}1}$*, namely*

$$\boldsymbol{\beta}^{\text{SVM}1} \begin{cases} \sim \mathbf{x}_1 & \text{if } f(\mathbf{x}_2; \boldsymbol{\beta}^{\text{SVM}1}) > 1 \\ \sim \mathbf{x}_2 & \text{if } f(\mathbf{x}_1; \boldsymbol{\beta}^{\text{SVM}1}) > 1 \\ \perp \mathbf{x}_2 - \mathbf{x}_1 & \text{otherwise, with } f(\mathbf{x}_1; \boldsymbol{\beta}^{\text{SVM}1}) = f(\mathbf{x}_2; \boldsymbol{\beta}^{\text{SVM}1}) = 1 \end{cases}$$

Thus, except in very special cases, the directions of vectors $\boldsymbol{\beta}_B^{cent}$ *and* $\boldsymbol{\beta}^{\text{SVM}1}$ *are unrelated. There are many different configurations of 3 points in* $\mathbb{R}^n$ *where the centroid vector* $\boldsymbol{\beta}_B^{cent}$ *lies in a different direction to the hinge-loss* SVM *solution*

$\boldsymbol{\beta}^{\text{SVM}1}$. *Figure 4.1 illustrates four such examples.*

Theorem 4.28 (Pointwise convergence). *If the kernel machine f_C is generated following one of the cases in Definition 4.26, then*

$$\lim_{C\to 0^+} \frac{f_C(\mathbf{x})}{C} = \begin{cases} \text{sign}(B) \times \infty & \text{if } \mathcal{R}_0 \text{ holds and } B \neq 0 \\ g_B(\mathbf{x}) & \text{if } \mathcal{R}_{\text{hom}} \text{ holds} \\ g_B(\mathbf{x}) + B & \text{if } \mathcal{R}_{bound} \text{ holds} \end{cases} \tag{4.8}$$

for every $\mathbf{x} \in \mathbb{R}^m$. In the $\mathcal{R}_0$ case:

$$\lim_{C\to 0^+} \frac{f_{H,C}(\mathbf{x})}{C} = (1 - B^2) g_0(\mathbf{x}) \tag{4.9}$$

for every $\mathbf{x} \in \mathbb{R}^m$.

Proof. Without loss of generality, assume that $k(\mathbf{x}, \mathbf{x}') := \langle \mathbf{x}, \mathbf{x}' \rangle$ with $\mathcal{H}$ isomorphic to $\mathbb{R}^m$. The hypothesis is then of the form

$$f(\mathbf{x}) = f_H(\mathbf{x}) + \mu_0 := \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0, \tag{4.10}$$

and $||f_H||^2_{\mathcal{H}} = ||\boldsymbol{\beta}||^2_2$ where $\boldsymbol{\beta} \in \mathbb{R}^m$. The proof is by considering each case ($\mathcal{R}_0$, $\mathcal{R}_{\text{hom}}$ and $\mathcal{R}_{\text{bound}}$ as defined in Definition 4.26) separately.

1. $\mathcal{R}_0$**:** First, consider the $\mathcal{R}_0$ case (4.5) for hypothesis (4.10) where

$$(\boldsymbol{\beta}_C, \mu_C) := \arg\min_{\boldsymbol{\beta}, \mu_0} R[\boldsymbol{\beta}, \mu_0]$$

and

$$R[\boldsymbol{\beta}, \mu_0] := C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1 + yB}{2|I_y|} L(y, \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0) + ||\boldsymbol{\beta}||^2_2.$$

This implies

$$||\boldsymbol{\beta}_C||^2_2 \leq \min_{\boldsymbol{\beta}, \mu_0} R[\boldsymbol{\beta}, \mu_0] \leq R[\mathbf{0}, 0] = C \tag{4.11}$$

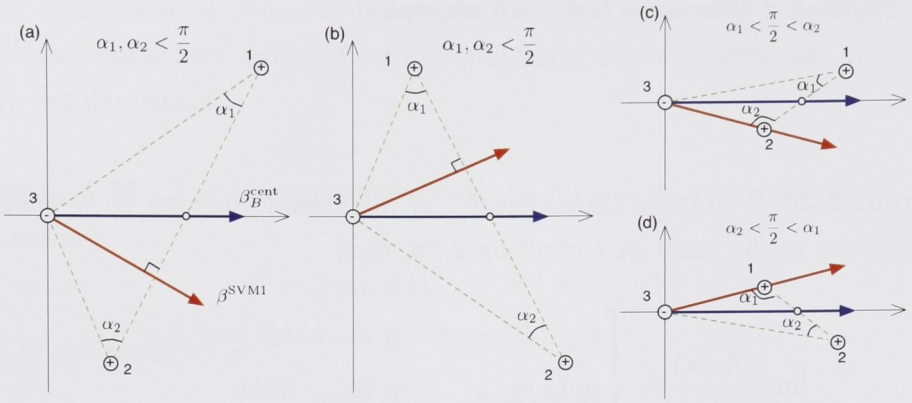


Figure 4.1: Example of difference between the hinge-loss and the centroid solutions. In each of the four cases, the centroid solution lies along the x-axis (the blue line), however the hinge-loss solution (red line) varies. In cases (a) and (b), the hinge-loss solution is perpendicular to the vector joining the two positive samples. In cases (c) and (d), the hinge-loss solution lies along one of the positive vectors.

as

$$\begin{aligned} R[\mathbf{0}, 0] &= \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} L(y,0) \\ &= \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} \\ &= \frac{1+B}{2} + \frac{1-B}{2} = 1. \end{aligned}$$

The loss functions considered in this theorem (see Definition 4.26) are continuously differentiable, thus the critical point equations must hold:

$$\frac{\partial}{\partial \mu_C} R[\boldsymbol{\beta}_C, \mu_C] = 0 \tag{4.12}$$

$$\frac{\partial}{\partial \boldsymbol{\beta}_C} R[\boldsymbol{\beta}_C, \mu_C] = 0 \tag{4.13}$$

(1.A) **The $L(y,\xi) = (1-y\xi)^2$ case:** Solving (4.12) gives

$$\begin{aligned}\frac{\partial}{\partial \mu_C} R[\boldsymbol{\beta}_C, \mu_C] &= -2C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle - y\mu_C) \\ &= -2C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}(y - \langle \mathbf{x}, \boldsymbol{\beta}_C\rangle - \mu_C) = 0 \\ \Rightarrow \mu_C &= \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}(y - \langle \mathbf{x}, \boldsymbol{\beta}_C\rangle) \qquad (4.14) \\ &= B - \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle \qquad (4.15)\end{aligned}$$

where (4.14) follows from

$$\sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} = 1,$$

and (4.15) from

$$\sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} y = B.$$

It follows from (4.11) that

$$\left| \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle \right| \leq ||\boldsymbol{\beta}_C||_2 \max_{\mathbf{x}\in\mathcal{X}} ||\mathbf{x}||_2 \leq \sqrt{C} \max_{\mathbf{x}\in\mathcal{X}} ||\mathbf{x}||_2.$$

Therefore,

$$\mu_C = B + O(\sqrt{C}) \qquad (4.16)$$

where $O(\xi)$ denotes a term such that $|O(\xi)/\xi|$ is bounded for $\xi > 0$.

Solving (4.13) gives

$$\begin{aligned}2\boldsymbol{\beta}_C - 2C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y(\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle + \mu_C))\mathbf{x} = 0 \\ \Rightarrow \boldsymbol{\beta}_C = C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y(\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle + \mu_C))\mathbf{x}.\end{aligned}$$

From (4.11), the bound $\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle \leq ||\mathbf{x}||_2 ||\boldsymbol{\beta}_C||_2 = O(\sqrt{C})$ holds, and

combined with (4.16) gives

$$\begin{aligned}\frac{f_{H,C}(\mathbf{x}')}{C} &= \frac{\langle \mathbf{x}', \boldsymbol{\beta}_C \rangle}{C} \\ &= \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{(1+yB)(1-yB+O(\sqrt{C}))}{2|I_y|} \langle \mathbf{x}, \mathbf{x}' \rangle \\ &= \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{(1-B^2+O(\sqrt{C}))}{2|I_y|} \langle \mathbf{x}, \mathbf{x}' \rangle \\ &= (1-B^2) g_0(\mathbf{x}') + O(\sqrt{C})\end{aligned}$$

and

$$\begin{aligned}\frac{f_C(\mathbf{x})}{C} &= \frac{f_{H,C}(\mathbf{x}) + \mu_C}{C} \\ &= (1-B^2) g_0(\mathbf{x}) + O(\sqrt{C}) + \frac{B + O(\sqrt{C})}{C} \\ &= (1-B^2) g_0(\mathbf{x}) + O(\sqrt{C})\end{aligned}$$

This proves the theorem for $\mathcal{R}_0$ in the least-squares loss case.

□

(1.B) **The** $L(y,\xi) = \max(0, 1-y\xi)^2$ **case:** It follows that

$$|f_C(x)| = |\langle \mathbf{x}, \boldsymbol{\beta}_C \rangle + \mu_C| \leq ||\boldsymbol{\beta}_C||_2 ||\mathbf{x}||_2 + |\mu_C|$$

for any $x \in \mathbb{X}$. Applying the bounds (4.11) and (4.16) yields

$$||\boldsymbol{\beta}_C||_2 ||\mathbf{x}||_2 + |\mu_C| \leq \sqrt{C} ||\mathbf{x}||_2 + |B| + O(\sqrt{C}) = |B| + O(\sqrt{C}).$$

By the $\mathcal{R}_0$ assumptions, $|B| < 1$ and therefore every sample becomes a support vector for sufficiently small C. Thus, this reduces to the previous case.

□

---

2. $\mathcal{R}_{\text{hom}}$**:** Consider now the $\mathcal{R}_{\text{hom}}$ case (4.6) for hypothesis (4.10) where

$$\boldsymbol{\beta}_C := \arg\min_{\boldsymbol{\beta}} R[\boldsymbol{\beta}]$$

and the regularised risk functional is

$$R[\boldsymbol{\beta}] := C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} L(y, \langle \mathbf{x}, \boldsymbol{\beta}\rangle) + ||\boldsymbol{\beta}||_2^2. \tag{4.17}$$

The bound

$$||\boldsymbol{\beta}_C||_2^2 \leq R[\mathbf{0}] = C \tag{4.18}$$

still holds in this case.

(2.A) **The $L(y,\xi) = (1-y\xi)^2$ case:** As in the previous $\mathcal{R}_0$ case, the critical point equation must hold:

$$\begin{aligned} \frac{\partial}{\partial\boldsymbol{\beta}} R[\boldsymbol{\beta}_C] &= 2\boldsymbol{\beta}_C - 2C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}(1 - y\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle)\mathbf{x} = 0 \\ \Rightarrow \boldsymbol{\beta}_C &= C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}(1 - y\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle)\mathbf{x}. \end{aligned}$$

Combining with the bound (4.18) yields

$$\begin{aligned} \frac{f_{H,C}(\mathbf{x}')}{C} &= \langle \mathbf{x}', \boldsymbol{\beta}_C\rangle \\ &= \sum_{(\mathbf{x},y)\in\mathcal{X}} y\frac{(1+yB)(1+O(\sqrt{C}))}{2|I_y|}\,\langle \mathbf{x}, \mathbf{x}'\rangle \\ &= \sum_{y\in\{1,-1\}} y\frac{1+yB}{2|I_y|} \sum_{\mathbf{x}\in\mathcal{X}_y} \langle \mathbf{x}, \mathbf{x}'\rangle + O(\sqrt{C}) \\ &= g_B(\mathbf{x}') + O(\sqrt{C}) \end{aligned}$$

where $\mathcal{X}_y := \{\mathbf{x}' | y = y' \;\&\; (\mathbf{x}', y') \in \mathcal{X}\}$. This proves the case of $\mathcal{R}_{\text{hom}}$ for least-squares loss.

□

(2.B) **The $L(y,\xi) = \max(0, 1-y\xi)^2$ case:** The bound (4.18) yields

$$|\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle\,| \leq ||\mathbf{x}||_2 ||\boldsymbol{\beta}_C||_2 = O(\sqrt{C}),$$

and thus every sample becomes a support vector for sufficiently small C. This reduces the problem to the previous case.

□

(2.C) **The $L(y, \xi) = \max(0, 1 - y\xi)$ case:** By the bound (4.18), every point becomes a support vector for sufficiently small C as in the previous case. Differentiating the risk functional (4.17) for this loss yields

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\beta}} R[\boldsymbol{\beta}_C] &= \boldsymbol{\beta}_C - C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}\mathbf{x} = 0 \\ \Rightarrow \boldsymbol{\beta}_C &= C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}\mathbf{x}.\end{aligned}$$

It follows that

$$\begin{aligned}\frac{f_{H,C}(\mathbf{x}')}{C} &= \langle \mathbf{x}', \boldsymbol{\beta}_C \rangle \\ &= \sum_{y\in\{1,-1\}} y \frac{1+yB}{2|I_y|} \sum_{\mathbf{x}\in\mathcal{X}_y} \langle \mathbf{x}, \mathbf{x}' \rangle \\ &= g_B(\mathbf{x}')\end{aligned}$$

□

---

3. **$\mathcal{R}_{\text{bound}}$:** Finally, consider the bounded risk case $\mathcal{R}_{\text{bound}}$ (4.7) of the hypothesis (4.10) where

$$(\boldsymbol{\beta}_C, \mu_C) := \arg\min_{\boldsymbol{\beta},\mu_0} R[\boldsymbol{\beta}, \mu_0]$$

and the regularised risk functional is

$$R[\boldsymbol{\beta}, \mu_0] := C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|} L(y, \langle \mathbf{x}, \boldsymbol{\beta} \rangle + \mu_0) + ||\boldsymbol{\beta}||_2^2 + \mu_0^2.$$

Again, the bound

$$||\boldsymbol{\beta}_C||_2^2 \leq R[\mathbf{0}, 0] = C$$

holds.

(3.A) **The $L(y, \xi) = (1 - y\xi)^2$ case:** Equating the partial derivatives of the

risk functional to zero gives

$$\begin{aligned}
\frac{\partial}{\partial \mu_C} R[\boldsymbol{\beta}_C, \mu_C] &= 2\mu_C - 2C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y(\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle + \mu_C)) \\
&= 0 \\
\Rightarrow \mu_C &= C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle) \\
&= CB - C \sum_{(\mathbf{x},y)\in\mathcal{X}} \frac{1+yB}{2|I_y|}\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle \\
&= CB + CO(\sqrt{C}) \\
&= O(C\sqrt{C})
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\beta}_C} R[\boldsymbol{\beta}_C, \mu_C] &= 2\boldsymbol{\beta}_C - 2C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle - \mu_C)\mathbf{x} \\
&= 0 \\
\Rightarrow \boldsymbol{\beta}_C &= C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}(1 - y\,\langle \mathbf{x}, \boldsymbol{\beta}_C\rangle - \mu_C)\mathbf{x}
\end{aligned}$$

The solutions now follow easily:

$$\begin{aligned}
\Rightarrow \frac{f_{H,C}(\mathbf{x}')}{C} &= \frac{\langle \mathbf{x}', \boldsymbol{\beta}_C\rangle}{C} \\
&= \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{(1+yB)(1+O(C\sqrt{C}))}{2|I_y|}\,\langle \mathbf{x}, \mathbf{x}'\rangle \\
&= \sum_{y\in\{1,-1\}} y \frac{1+yB}{2|I_y|} \sum_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{x}, \mathbf{x}'\rangle + O(C\sqrt{C}) \\
&= g_B(\mathbf{x}') + O(C\sqrt{C}),
\end{aligned}$$

and

$$\begin{aligned}
\frac{f_C(\mathbf{x}')}{C} &= \frac{f_{H,C}(\mathbf{x}') + \mu_C}{C} \\
&= g_B(\mathbf{x}') + O(C\sqrt{C}) + \frac{CB + CO(\sqrt{C})}{C} \\
&= g_B(\mathbf{x}') + B + O(C\sqrt{C}).
\end{aligned}$$

□

(3.B) **The $L(y,\xi) = \max(0, 1-y\xi)^2$ case:** Again, this reduces to the previous case as every sample becomes a support vector for sufficiently small C.

□

(3.C) **The $L(y,\xi) = \max(0, 1-y\xi)$ case:** Every sample becomes a support vector for sufficiently small C. Equating the partial derivative with respect to $\boldsymbol{\beta}$ of the risk to zero gives

$$\frac{\partial}{\partial \boldsymbol{\beta}} R[\boldsymbol{\beta}_C] = \boldsymbol{\beta}_C - C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}\mathbf{x} = 0$$
$$\Rightarrow \boldsymbol{\beta}_C = C \sum_{(\mathbf{x},y)\in\mathcal{X}} y \frac{1+yB}{2|I_y|}\mathbf{x}.$$

It follows that

$$\begin{aligned}\frac{f_{H,C}(\mathbf{x}')}{C} &= \sum_{y\in\{1,-1\}} y \frac{1+yB}{2|I_y|} \sum_{\mathbf{x}\in\mathcal{X}_y} \langle \mathbf{x}, \mathbf{x}' \rangle \\ &= g_B(\mathbf{x}') \\ \frac{f_C(\mathbf{x}')}{C} &= g_B(\mathbf{x}') + \frac{CB + CO(\sqrt{C})}{C} \\ &= g_B(\mathbf{x}') + B + O(\sqrt{C})\end{aligned}$$

□

□

4.4 Convergence of Performance Metrics

The pointwise convergence proof of the previous section is not sufficient to guarantee the convergence of some performance measures. In the case of continuous measures, a straightforward extension of Theorem 4.28 guarantees the convergence in the high-regularisation limit. However, this does not apply for non-continuous measures, which are more common than continuous metrics. This section examines the convergence of non-continuous measures, based on Kowalczyk (2007b). The main proof of convergence uses the transversality theory presented in Section 4.1.

Definition 4.29 (Thin definitions). *A subset $A \subset \mathbb{R}^m$ is negligible if it has a Lebesgue measure of 0.*

A hypothesis $f\colon \mathbb{R}^m \to \mathbb{R}$ is called a thin hypothesis if $f^{-1}[\nu]$ is negligible for all $\nu \in \mathbb{R}$.

A feature map $\Phi\colon \mathbb{R}^m \to \mathcal{H}$ is called thin if every hypothesis

$$f(\mathbf{x}) = \langle \Phi(\mathbf{x}), \boldsymbol{\beta} \rangle + \mu_0 \not\equiv const.$$

is thin.

Finally, a kernel $k\colon \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is called thin if every admitted hypothesis

$$f(\mathbf{x}) = \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + \mu_0 \not\equiv const$$

is thin.

The first theorem shows that malicious examples can be constructed for which the pointwise convergence results hold, but the corresponding limits for the balanced error rate, error rate, and AROC do not.

Theorem 4.30. *Let $\mathcal{X}_{train}$ and $\mathcal{X}_{test}$ be training and testing datasets drawn from a Lebesgue measurable probability density on $\mathbb{R}^m \times \{1, -1\}$, and f_C be induced on $\mathcal{X}_{train}$. There exists a non-thin kernel $k \in C^\infty(\mathbb{R}^m \times \mathbb{R}^m) \to \mathbb{R}$ such that for metric $\rho \in \{\text{AROC}, \text{err}, \text{balerr}\}$*

$$\lim_{C \to 0^+} \rho(\mathcal{X}_{test}, \frac{f_C}{C}) \neq \rho(\mathcal{X}_{test}, \lim_{C \to 0^+} \frac{f_C}{C})$$

in the $\mathcal{R}_{\text{hom}}$ or $\mathcal{R}_{bound}$ cases, and

$$\lim_{C \to 0^+} \rho(\mathcal{X}_{test}, \frac{f_{H,C}}{C}) \neq \rho(\mathcal{X}_{test}, \lim_{C \to 0^+} \frac{f_{H,C}}{C})$$

in the $\mathcal{R}_0$ case.

Before proving this theorem, the following lemma is needed. It proves that the existence of a smooth map from any disjoint closed subsets to a finite set of points. This lemma allows the simplification of the theorem's proof to a specific set of points in two-dimensions.

Lemma 4.31. *For any finite set of points $\{\mathbf{v}_i\}_{i \in I} \subset \mathbb{R}^n$ and for any disjoint closed subsets $\{V_i \subset \mathbb{R}^m\}_{i \in I}$, there exists a C^∞ mapping $\Phi\colon \mathbb{R}^m \to \mathbb{R}^n$ such that $V_i \subset \Phi^{-1}[\mathbf{v}_i]$.*

Proof. Let $\mathbf{v}_i \in \mathbb{R}^n$ and $V_i \subset \mathbb{R}^m$ for $i \in I$ be as in the theorem statement, and $\{\{U_{ij}\}_{j\in J_i}\}_{i\in I}$ be a set of open covers such that $\{U_{ij}\}_{j\in J_i}$ covers V_i and

$$\bigcap_{i\in I}\bigcup_{j\in J_i} U_{ij} = \emptyset.$$

Such a set exists as V_i is closed and disjoint. Let $\{\{g_{ik}\}_{k\in K_i}\}_{i\in I}$ be a set of smooth partitions of unity such that $\{g_{ik}\}_{k\in K_i}$ is subordinate to the cover $\{U_{ij}\}_{j\in J_i}$, and define

$$g_i := \sum_{k\in K_i} g_{ik}.$$

The partitions of unity are guaranteed to exist by Theorem 4.19. Then

$$\Phi := \sum_i \mathbf{v}_i g_i.$$

is smooth with inverse image $\Phi^{-1}[\mathbf{v}_i] \supset V_i$. □

The proof of Theorem 4.30 now follows.

Proof of Theorem 4.30. First consider the two dimensional case where

$$\mathcal{X}_{\text{train}} = \left\{ \left(\begin{bmatrix} -1 \\ 1 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} -2 \\ -1 \end{bmatrix}, -1 \right), \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 2 \\ -2 \end{bmatrix}, 1 \right) \right\}$$

is the training set, and

$$\mathcal{X}_{\text{test}} = \left\{ \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, 1 \right), \left(\begin{bmatrix} 0 \\ -1 \end{bmatrix}, -1 \right) \right\}$$

is the test set, and without loss of generality assume $C = C_+ = C_-$. Let X be the data matrix with rows $\mathbf{x}_i$ as previously. As the data points are centred by construction, the explicit ridge regression solution (see Definition 2.21) for $C = 1$ is

$$\boldsymbol{\beta}_1 = (X^*X + I)^{-1}X^*\mathbf{y} = \begin{bmatrix} .55 \\ .05 \end{bmatrix}.$$

Thus, $f_C(\mathbf{x}) \in \{0.05, -0.05\}$ for $\mathbf{x} \in \mathcal{X}_{\text{test}}$, and

$$\text{balerr}(\mathcal{X}_{\text{test}}, \text{sign} \circ f_C) = \text{err}(\mathcal{X}_{\text{test}}, \text{sign} \circ f_C) = 1 - \text{AROC}(\mathcal{X}_{\text{test}}, f_C) = 0.$$

By Theorem 4.28,

$$\lim_{C\to 0} f_C(\mathbf{x}) = g_B(\mathbf{x}) = g_0(\mathbf{x}) = 0$$

for all $\mathbf{x} \in \mathcal{X}_{\text{test}}$. It follows that

$$\text{balerr}(\mathcal{X}_{\text{test}}, \text{sign} \circ g_0) = \text{err}(\mathcal{X}_{\text{test}}, \text{sign} \circ g_0) = 1, \ \text{AROC}(\mathcal{X}_{\text{test}}, g_0) = 0.5$$

and therefore

$$\lim_{C\to 0^+} \rho(\mathcal{X}_{\text{test}}, \frac{f_C}{C}) \neq \rho(\mathcal{X}_{\text{test}}, \lim_{C\to 0^+} \frac{f_C}{C}).$$

As every sample becomes a support vector for sufficiently small C this proves the theorem for the $\mathcal{R}_0$ case and also covers the quadratic loss situations of $\mathcal{R}_{\text{hom}}$ and $\mathcal{R}_{\text{bound}}$. For the linear loss case of $\mathcal{R}_{\text{hom}}$

$$\boldsymbol{\beta}_1 = \begin{bmatrix} 1.54 \\ .39 \end{bmatrix},$$

thus

$$\text{balerr}(\mathcal{X}_{\text{test}}, \text{sign} \circ f_{H,C}) = \text{err}(\mathcal{X}_{\text{test}}, \text{sign} \circ f_{H,C}) = 1 - \text{AROC}(\mathcal{X}_{\text{test}}, f_{H,C}) = 0$$

and

$$\lim_{C\to 0^+} \rho(\mathcal{X}_{\text{test}}, \frac{f_{H,C}}{C}) \neq \rho(\mathcal{X}_{\text{test}}, \lim_{C\to 0^+} \frac{f_{H,C}}{C}).$$

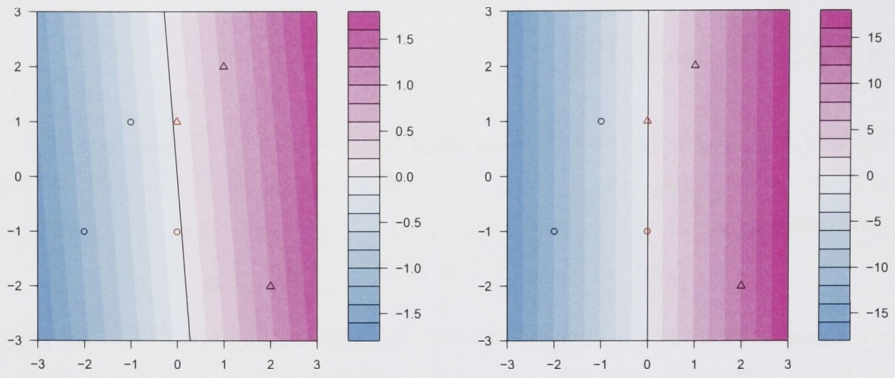
Finally, as the data is centred and $\mu_0 = 0$, this solution is also the $\mathcal{R}_{\text{bound}}$ solution. This proves the theorem in the $\mathcal{R}_{\text{hom}}$ and $\mathcal{R}_{\text{bound}}$ cases.

Generalisation to more complex cases is straightforward. For any Lebesgue measurable probability density on $\mathbb{X}$ there exists a C^∞ mapping $\Phi\colon \mathbb{R}^m \to \mathbb{R}^2$ that maps the dataset to the points in the 6-point example above by Lemma 4.31. Defining the kernel as $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ completes the proof. □

The specific set of points used in the proof of Theorem 4.30 and the resulting hyperplanes is illustrated in Figure 4.2.

The aim of the remainder of this section is to show that the inverse image of a separating linear hyperplane is negligible under some conditions. Consequently, the performance metrics almost surely converge in the limit of high regularisation, as the next theorem formally states.

Theorem 4.32. *Assume $\mathcal{X}_{train}$ and $\mathcal{X}_{test}$ are training and testing datasets drawn from a Lebesgue measurable probability density P on $\mathbb{R}^m \times \{1, -1\}$, and f_C be the hypothesis obtained from $\mathcal{X}_{train}$ using regularisation hyperparameter C (following*



(a) Ridge solution – C=1 (b) Centroid solution

Figure 4.2: Example ridge and centroid solutions. Red points indicate testing points. Black line indicates separating hyperplane obtained by training on the black points. With this particular choice of points, a metric measured on the ridge solution does not converge to the metric measured on the centroid solution in the limit of high regularisation.

the notation of the previous section). If $k \in C^\infty(\mathbb{R}^m \times \mathbb{R}^m, \mathbb{R})$ *is a thin kernel then for metric* $\rho \in \{\text{AROC}, \text{err}, \text{balerr}\}$

$$\lim_{C\to 0^+} \rho\left(\mathcal{X}_{test}, \frac{f_C}{C}\right) = \rho\left(\mathcal{X}_{test}, \lim_{C\to 0^+} \frac{f_C}{C}\right)$$

in the $\mathcal{R}_{\text{hom}}$ *and* $\mathcal{R}_{bound}$ *cases, and*

$$\lim_{C\to 0^+} \rho\left(\mathcal{X}_{test}, \frac{f_{H,C}}{C}\right) = \rho\left(\mathcal{X}_{test}, \lim_{C\to 0^+} \frac{f_{H,C}}{C}\right)$$

in the $\mathcal{R}_0$ *case.*

Proof. For the $\mathcal{R}_{\text{hom}}$ or $\mathcal{R}_{\text{bound}}$ cases and all $\rho \in \{\text{AROC}, \text{err}, \text{balerr}\}$, the following bound always holds:

$$\left|\lim_{C\to 0^+} \rho(\mathcal{X}_{\text{test}}, \frac{f_C}{C}) - \rho(\mathcal{X}_{\text{test}}, \lim_{C\to 0^+} \frac{f_C}{C})\right| \leq P(f(\mathbf{x}) = f(\mathbf{x}')|(\mathbf{x}, \cdot), (\mathbf{x}', \cdot) \in \mathcal{X}_{\text{test}}).$$

As k is a thin kernel, $P(f(\mathbf{x}) = f(\mathbf{x}')|(\mathbf{x}, \cdot), (\mathbf{x}', \cdot) \in \mathcal{X}_{\text{test}}) = 0$, hence

$$\lim_{C\to 0^+} \rho\left(\mathcal{X}_{\text{test}}, \frac{f_C}{C}\right) = \rho\left(\mathcal{X}_{\text{test}}, \lim_{C\to 0^+} \frac{f_C}{C}\right).$$

The proof for the $\mathcal{R}_0$ case follows similarly. □

Given this theorem, proving convergence of the performance metrics is reduced to showing the kernel or feature map is smooth and thin. Before showing that analytic hypothesis are thin, the next lemma is needed that proves the set difference between a k-jet fibre (see Section 4.1.5) and the fibre of a C^k hypothesis is negligible.

Lemma 4.33. *Let $f\colon U \to \mathbb{R}$ be a C^k hypothesis for $k \geq 1$. The set $f^{-1}[0] \setminus (j^k_* f)^{-1}[0]$ is negligible.*

Proof. The proof is by induction. Suppose $k = 1$. Choose $\mathbf{x}_0 \in f^{-1}[0] \setminus (j^1_* f)^{-1}[0]$ such that df is non-zero. By the inverse function theorem (Theorem 4.7), there exists $U_{\mathbf{x}_0} \subset U$ containing $\mathbf{x}_0$, $V_{\mathbf{x}_0} \subset \mathbb{R}$ containing $f(\mathbf{x}_0)$, and a map $g\colon V_{\mathbf{x}_0} \to U_{\mathbf{x}_0}$ such that $g(f(\mathbf{x} \in U_{\mathbf{x}_0})) = \mathbf{x}$. Consider the submanifold defined by $U' = f^{-1}[0] \cap U_{\mathbf{x}_0}$. As $f(\mathbf{x} \subset U') = 0$ can be solved for at least one dimension (by choice of $\mathbf{x}_0$), $\operatorname{codim}(U') \geq 1$ and U' is thus negligible. Assume true for $2, \ldots, k-1$. Write

$$
\begin{aligned}
f^{-1}[0] &= (f^{-1}[0] \setminus (j^{k-1}_* f)^{-1}[0]) \cup (j^{k-1}_* f)^{-1}[0] \\
&= (f^{-1}[0] \setminus (j^{k-1}_* f)^{-1}[0]) \cup (j^k_* f)^{-1}[0] \\
&\quad \cup \left(\bigcup_{|\boldsymbol{\alpha}|=k-1} (\partial_{\boldsymbol{\alpha}} f)^{-1}[0] \setminus (j^1_* \partial_{\boldsymbol{\alpha}} f)^{-1}[0] \right) \\
\Rightarrow f^{-1}[0] \setminus (j^k_* f)^{-1}[0] &= (f^{-1}[0] \setminus (j^{k-1}_* f)^{-1}[0]) \\
&\quad \cup \left(\bigcup_{|\boldsymbol{\alpha}|=k-1} (\partial_{\boldsymbol{\alpha}} f)^{-1}[0] \setminus (j^1_* \partial_{\boldsymbol{\alpha}} f)^{-1}[0] \right)
\end{aligned}
\tag{4.19}
$$

The first term of (4.19) is negligible by the inductive assumption. Likewise, the union of the second term is negligible as it is a countable union of negligible sets by the inductive assumption. As both terms are negligible, $f^{-1}[0] \setminus (j^k_* f)^{-1}[0]$ is negligible. □

Theorem 4.34. *Non-constant C^ω hypotheses on a connected domain are thin.*

Proof. Let $U \subset \mathbb{R}^n$ be an open connected subset and $f\colon U \to \mathbb{R}$ be a C^ω hypothesis. Choose $A \subset U$ as a compact and connected subset. The set U is the union of a countable number of such sets, thus it is sufficient to prove that either $f[A] \equiv const$ or $A \cap f^{-1}[0]$ is negligible.

Consider the sequence of nested closed subsets

$$A \supset A \cap (j^1_* f)^{-1}[0] \supset A \cap (j^2_* f)^{-1}[0] \supset \cdots \supset A \cap (j^k_* f)^{-1}[0] \cap \cdots$$

Suppose the sequence does not terminate. Then, there exists

$$A' := \bigcap_{k=1}^{\infty} (j^k_* f)^{-1}[0] \cap A \neq \emptyset.$$

As f is analytic, it follows that for any $x_0 \in A'$, $(j^\infty_{x_0} f)(z) \equiv 0$ for z sufficiently close to x_0 and therefore A' is closed and open. Consequently, as U is connected and A' is non-empty, $A' = U$ and $f \equiv const$ over the entire domain U. This contradicts the assumption that f is non-constant, thus the sequence must terminate.

It follows that there exists k_0 such that $A \cap (j^k_* f)^{-1}[0] = \emptyset$ for all $k \geq k_0$. Then

$$A \cap f^{-1}[0] = A \cap f^{-1}[0] \setminus (j^k_* f)^{-1}[0] \subset f^{-1}[0] \setminus (j^k_* f)^{-1}[0].$$

The last term is negligible by Lemma 4.33 thus $A \cap f^{-1}[0]$ is negligible. □

Corollary 4.35. *Analytic kernels are thin.*

Proof. This follows immediately from Theorem 4.34 and Definition 4.29. □

It follows from Corollary 4.35 that the popular polynomial and Gaussian RBF kernels are thin as they are analytic. Consequently, the performance metric (AROC, err, etc) in the high-regularisation SVM limit with a polynomial or RBF kernel is equivalent to the centroid projection.

Theorem 4.36. *Let $U \subset \mathbb{R}^n$ be an open and connected subset, $\Phi \in C^k(U, \mathbb{R}^m)$ be a feature map, and $d^k_n \geq m + n$ where d^k_n is the dimension of the polynomial vector space of degree $\leq k$ in n-variables. The set of polynomials $\{p | p \in P^k(\mathbb{R}^n, \mathbb{R}^m)$ & $\Phi + p$ is thin$\}$ is a residual set with negligible complement.*

Proof. Let the assumptions of the theorem hold, $\Phi = (\phi_i)$, and $p = (p_i)$. Denote the hypothesis as

$$f_{\Phi,\boldsymbol{\beta},\mu_0}(\mathbf{x}) := \langle \Phi(\mathbf{x}), \boldsymbol{\beta} \rangle + \mu_0$$

for $\boldsymbol{\beta} \in \mathbb{R}^m$ and $\mu_0 \in \mathbb{R}$. To show that $f_{\Phi+p,\boldsymbol{\beta},\mu_0}(\mathbf{x})$ is thin, it is necessary to show that $f^{-1}_{\Phi+p,\boldsymbol{\beta},\mu_0}[\nu]$ is negligible for all $\nu \in \mathbb{R}$. As ν can be absorbed into the bias, it is therefore sufficient to show that $f^{-1}_{\Phi+p,\boldsymbol{\beta},\mu_0}[0]$ is negligible for all $\mu_0 \in \mathbb{R}$.

From Lemma 4.33

$$f^{-1}_{\Phi+p,\boldsymbol{\beta},\mu_0}[0] \setminus (j^k_* f_{\Phi+p,\boldsymbol{\beta},\mu_0})^{-1}[0]$$

is negligible, thus it is sufficient to show that for sufficiently large k,

$$(j^k_* f_{\Phi+p,\boldsymbol{\beta},\mu_0})^{-1}[0] = \emptyset,$$

or equivalently

$$\left\langle (j^k_*(\Phi+p))(\mathbf{x}), \boldsymbol{\beta} \right\rangle + \mu_0 \neq 0 \in J^k(U, \mathbb{R}^m)$$

for all $0 \neq \boldsymbol{\beta} \in \mathbb{R}^m$, $\mu_0 \in \mathbb{R}$, and $\mathbf{x} \in U$. Decomposing the first term into the homogenous portion plus the degree 0 term gives

$$\left\langle (j^k_*(\Phi+p))(\mathbf{x}), \boldsymbol{\beta} \right\rangle + \mu_0 = \underbrace{\left\langle (j^k_H(\Phi+p))(\mathbf{x}), \boldsymbol{\beta} \right\rangle}_{\in P^k_H(U,\mathbb{R}^m)} + \underbrace{\langle (\Phi+p)(\mathbf{x}), \boldsymbol{\beta} \rangle + \mu_0}_{\in \mathbb{R}} \neq 0,$$

where j^k_H denotes the homogenous portion of j^k_* and $P^k_H(U, \mathbb{R}^m)$ denotes the homogenous polynomial vector space, which is isomorphic to $\mathbb{R}^{(d^k_n - 1)m}$. As this condition must hold for all $\mu_0 \in \mathbb{R}$, it reduces to

$$\left\langle (j^k_H(\Phi+p))(\mathbf{x}), \boldsymbol{\beta} \right\rangle \neq 0,$$

or equivalently

$$\operatorname{rank}[j^k_H(\phi_1+p_1)(\mathbf{x}), \ldots, j^k_H(\phi_m+p_m)(\mathbf{x})] = m \tag{4.20}$$

for all $\mathbf{x} \in U$, where rank denotes the matrix rank.

Let

$$W := \left\{ (\mathbf{u}, (\mathbf{v}_i)) \in \mathbb{R}^{n+m} \times \left(\mathbb{R}^{d^k_n - 1}\right)^m = J^k(\mathbb{R}^n, \mathbb{R}^m) \middle| \operatorname{rank}[\mathbf{v}_1, \ldots, \mathbf{v}_m] < m \right\}. \tag{4.21}$$

Due to the condition (4.20), it is sufficient to show that $(j^k(\Phi+p))(\mathbf{x}) \notin W$. Defining

$$W_q := \{(\mathbf{u}, (\mathbf{v}_i)) \in W \,|\operatorname{rank}[\mathbf{v}_1, \ldots, \mathbf{v}_m] = q\}$$

yields a sequence such that $W = \bigcup_{q=1}^{m-1} W_q$. Furthermore, as a $(d^k_n - 1) \times m$ matrix of rank q is a submanifold with codimension $(d^k_n - 1 - q)(m - q)$ (Demazure, 2000,

Chapter 2.2), W_q is a submanifold with

$$\begin{aligned}\operatorname{codim}(W_q) &= (d_n^k - 1 - q)(m - q) \\ &\geq d_n^k - m \\ &\geq n\end{aligned}$$

by the theorem assumption $d_n^k \geq m + n$.

Using Thom's transversality theorem (see Theorem 4.24), the set S of polynomials $p \in P^k(\mathbb{R}^n, \mathbb{R}^m)$ such that $(j^k(\Phi + p))(\mathbf{x})$ is transverse to W_q for all $q \in \{1, \ldots, m-1\}$ is a residual set with *negligible complement*, thus only the polynomials in set S need to be considered. Recall that transversality implies

$$\operatorname{codim}(W_q) + \operatorname{codim}((j^k(\Phi + p))[U]) = \operatorname{codim}(W_q \cap (j^k(\Phi + p))[U]).$$

As $\operatorname{codim}(W_q) \geq n$, it follows that

$$W_q \cap (j^k(\Phi + p))[U] = \emptyset$$

for all $q \in \{1, 2, \ldots, m-1\}$. Therefore, $(j^k(\Phi + p))(\mathbf{x}) \notin W$ for all $p \in S$ and $\mathbf{x} \in U$. □

The consequences of this theorem is that any non-thin C^k map becomes thin under a small polynomial perturbation.

4.5 Fast Estimation of Generalisation Error

Due to the simplicity of the centroid classifier, optimisations can be made to reduce the computational cost of resampling methods for estimating the generalisation error. More specifically, given a hypothesis induced from the whole dataset, it is computationally cheap to update the hypothesis to reflect the removal of a small subset of samples. More specifically, leave-one-out (LOO) and leave-two-out resampling estimates (see Section 2.6) for AROC and err can be calculated analytically.

Let $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{X} \times \{1, -1\}$ be a finite dataset. Using a linear homogeneous hypothesis $f(\mathbf{x}) := \langle \mathbf{x}, \boldsymbol{\beta} \rangle$, the centroid solution with $B = 0$ is given by

$$\boldsymbol{\beta} = \frac{\boldsymbol{\beta}_+}{|I_+|} - \frac{\boldsymbol{\beta}_-}{|I_-|}$$

where $\boldsymbol{\beta}_+ = \sum_{i \in I_+} \mathbf{x}_i$, $\boldsymbol{\beta}_- = \sum_{i \in I_-} \mathbf{x}_i$, $I_+ = \{i | y_i = 1\}$, and $I_- = \{i | y_i = -1\}$. Let $L \colon \{1, -1\} \times \mathbb{R} \to \mathbb{R}$ be a loss function, and $y_i' := \frac{y_i + 1}{2}$. The LOO error is then

$$\mathrm{err}_{\mathrm{LOO}} = \frac{1}{n} \sum_{i \in I} L \left(y_i, \left\langle \frac{\boldsymbol{\beta}_+ - y_i' \mathbf{x}_i}{|I_+| - y_i'} - \frac{\boldsymbol{\beta}_- - (1 - y_i') \mathbf{x}_i}{|I_-| - (1 - y_i')}, \mathbf{x}_i \right\rangle \right).$$

The LTO estimate of AROC is also easily calculable, though has a higher computational cost:

$$\mathrm{AROC}_{\mathrm{LTO}} = \frac{1}{|I_+||I_-|} \sum_{i \in I_+} \sum_{j \in I_-} L_{\mathrm{LTO}} \left(\left\langle \hat{\boldsymbol{\beta}}_{ij}, \mathbf{x}_i \right\rangle, \left\langle \hat{\boldsymbol{\beta}}_{ij}, \mathbf{x}_j \right\rangle \right)$$

where

$$\hat{\boldsymbol{\beta}}_{ij} := \frac{\boldsymbol{\beta}_+ - \mathbf{x}_i}{|I_+| - 1} - \frac{\boldsymbol{\beta}_- - \mathbf{x}_j}{|I_-| - 1}.$$

and

$$L_{\mathrm{LTO}}(y, y') := \begin{cases} 1 & \text{if } y > y' \\ 0.5 & \text{if } y = y' \\ 0 & \text{otherwise} \end{cases}.$$

As the LTO estimate of AROC involves a sum over all pairs it scales with $O(n^2)$, and thus is only practical for small datasets.

4.6 Recursive Feature Elimination

The concept of recursive feature elimination (RFE) as an embedded feature selection method has already been introduced (Section 2.5). Here, the application of it to the quadratic loss SVM, hereby known as the *recursive feature elimination support vector machine* (RFE–SVM), is studied. The RFE–SVM was originally proposed by Guyon et al. (2002) and, despite the moderate computational cost, has been used to analyse microarray data (Alon et al., 1999; Ambroise and McLachlan, 2002; Huang and Kecman, 2005). Huang and Kecman (2005) made the observation that increasing the amount of regularisation for the quadratic loss RFE–SVM increased the performance of the algorithm on the datasets used. By Theorem 4.28, the high-regularisation limit of the SVM is the centroid solution, thus one can apply RFE to the centroid and obtain RFE–SVM in the limit of high-regularisation. Furthermore, as the feature weights β_i are independent, the retraining step after the elimination of features is unnecessary and the RFE

reduces to a simple ranking filter method (Section 2.5) with

$$v_j = \frac{1}{|I_+|} \sum_{i \in I_+} x_{ij} - \frac{1}{|I_-|} \sum_{i \in I_-} x_{ij}.$$

This filter is called *centroid feature selection* (CFS), with the combination of CFS and the centroid classifier called CFSCentroid.

This selection method, like the centroid classifier itself, is related to MMD and the *Hilbert–Schmidt independence criterion* (HSIC); it is equivalent to performing supervised feature selection using a linear kernel and the HSIC with appropriate scaling of the dataset. An evaluation of the HSIC selection approach with various kernels against several alternative methods was published by (Song et al., 2007b,a), but the results are not presented in this thesis.

4.7 Empirical Analysis

4.7.1 Comparison against the RFE–SVM

The centroid and RFE–SVM solutions were compared on a variety of different bioinformatics datasets. The centroid classifier used in experiments was the linear centroid projection defined in Definition 4.25 with $B = 0$ and the bias μ_0 chosen to place the decision hyperplane equidistant from the two class centres:

$$\mu_0 = -\left\langle \frac{\bar{\mathbf{x}}_+ + \bar{\mathbf{x}}_-}{2}, \boldsymbol{\beta} \right\rangle = \frac{||\bar{\mathbf{x}}_+||_2^2 - ||\bar{\mathbf{x}}_-||_2^2}{4}$$

where $\bar{\mathbf{x}}_+ := \frac{1}{|I_+|} \sum_{i \in I_+} \mathbf{x}_i$ and $\bar{\mathbf{x}}_- := \frac{1}{|I_-|} \sum_{i \in I_-} \mathbf{x}_i$. The regularisation constant was evaluated at two different values ($\frac{1}{C} = \lambda = \{.01, 100\}$) to observe the effect of both large and small amounts of regularisation. The bootstrap (Section 2.6) was chosen to evaluate the generalisation performance of the models.

The first three datasets evaluated – *colon*, *van 't Veer*, and *lymphoma* – are popular freely available "benchmark" datasets. The first, a colon cancer dataset (Alon et al., 1999; Ambroise and McLachlan, 2002; Guyon et al., 2002; Huang and Kecman, 2005), is a two-class classification problem with 62 samples (22 normal and 40 cancerous) and 2000 dimensions. The second is a two-class breast cancer dataset (van 't Veer et al., 2002) consisting of 98 samples, 46 with a distant metastasis and 52 with no metastasis. Each sample has 5952 dimensions. The final is a multi-class lymphoma dataset with 62 samples – 11 chronic lymphocytic

leukaemia (CLL), 42 diffuse large-cell lymphoma (DLCL), 9 follicular lymphoma (FL) – each sample with 4026 dimensions.

The .632+ bootstrap results of the AROC and balanced error rate metrics for these three datasets are shown in Figure 4.3 to Figure 4.5, with full bootstrap results available in Appendix B. The error bars in the figure are 95% confidence intervals. The colon results (Figure 4.3) show that increasing the amount of regularisation increases the performance of the RFE–SVM, as was observed by Huang and Kecman (2005). The centroid shown here is clearly equivalent in performance to the high-regularisation RFE–SVM using the AROC metric, but is of course considerably quicker. The difference between the RFE–SVM with $\lambda = 100$ and centroid method using balanced error arises due to the positioning of the centroid hyperplane to be equidistant to the class centroids; the AROC is sensitive only to orientation not positioning.

The next dataset was originally analysed by van 't Veer et al. (2002). The dataset was reanalysed with RFE–SVM and with the centroid classifier producing the results in Figure 4.4. On this dataset, the RFE–SVM with high regularisation parameter $\lambda = 100$ produced the best result, though the centroid is close to the same level of performance and achieves similar AROC performance.

The lymphoma dataset is a multiclass dataset unlike the colon and van 't Veer datasets. As the centroid projection and SVM are two-class classifiers, the *one-vs-all* was used for the multiclass classification. Again, the same analysis was repeated and the results shown in Figure 4.5 obtained. The difference between the centroid and RFE–SVM solutions is striking – centroid achieves near perfect classification at 1024 features while the RFE–SVM performs significantly worse at the same number of features. The RFE–SVM achieves its best performance of just under 5% balanced error and just under 98% AROC at 8 features, but this level is far from the best achieved by the centroid method.

The final dataset is a multiclass dataset with 11 different classes and 186 samples with the smallest class containing 6 samples. Again, the OVA architecture was used to construct a multiclass classifier. Unlike the previous datasets, this is a quantitative polymerase chain reaction (QPCR) dataset not a microarray dataset. QPCR is considered a higher-quality technique, though more costly and with considerably less markers available. Each of the 186 samples in the dataset contain QPCR measurements for 740 markers. This dataset is an unpublished dataset created from results of the initial dataset studied by Tothill et al. (2005). Figure 4.6 shows the results of the same analysis on this dataset. Using the

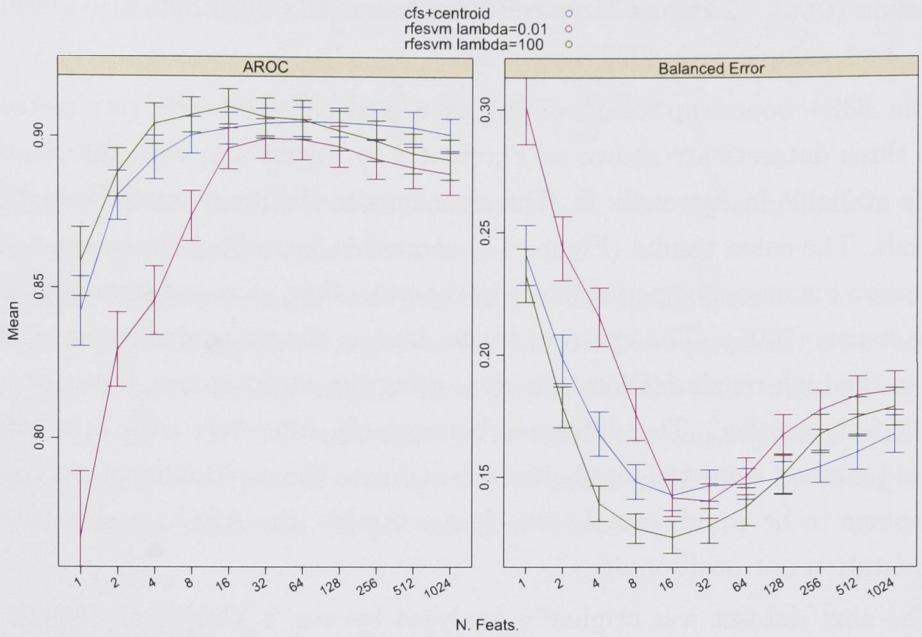


Figure 4.3: .632+ Bootstrap results for the centroid and RFE–SVM classifiers on the *colon* cancer dataset. Error bars are 95% confidence intervals. There is no statistical difference between the RFE–SVM and centroid classifiers at $\lambda = 100$ using the AROC metric. The small difference observable in the balanced error arises due to the positioning of the centroid hyperplane to be equidistant to the class centroids; the AROC is sensitive only to orientation not positioning.

AROC metric, the centroid and high-regularisation RFE–SVM ($\lambda = 100$) perform equivalently with the low-regularisation RFE–SVM ($\lambda = .01$) performing worse. The balanced error rate results show the centroid performing equivalently to the high-regularisation RFE–SVM for sufficiently small sets of features, but both RFE–SVM solutions perform better with large numbers of features. The low-regularisation RFE–SVM performs drastically worse than its high-regularisation counterpart. As lower numbers of features are desirable for this project, the centroid solution is a good – and significantly faster – alternative to the full high-regularisation RFE–SVM solution.

4.7.2 Comparison against Shrunken Centroid

Tibshirani et al. (2002, 2003) introduced a different centroid based classification and selection procedure targeted to microarray data known as the *shrunken centroid* or PAM classifier. It is a multiclass classifier with incorporated feature selection method. Feature selection is carried out through shrinking the individual

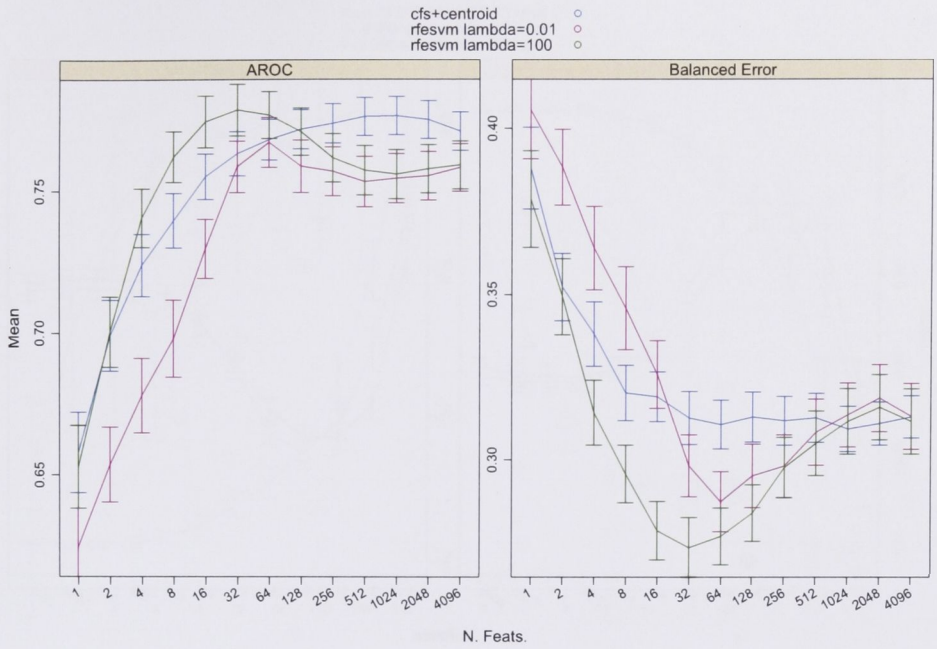


Figure 4.4: .632+ Bootstrap results for the centroid and RFE–SVM classifiers on the *van 't Veer* dataset. Error bars are 95% confidence intervals.

class centroids towards the overall centroid. The amount of shrinkage determines the number of genes, with all genes being eliminated when each shrunken class centroid equals the overall centroid.

Let

$$\bar{\mathbf{x}}_k = (\bar{x}_{kj})^m := E[\mathbf{x}|y = k]$$

be the class centroid for class $k \in \mathbb{Y}$ and

$$\bar{\mathbf{x}} = (\bar{x}_j)^m := E[\mathbf{x}]$$

be the overall centroid. Define the t-statistic

$$\mathbf{d}_k = (d_{kj})^m := \frac{\bar{\mathbf{x}}_k - \bar{\mathbf{x}}}{m_k(\sigma + \sigma_0)}$$

$$\Rightarrow d_{kj} = \frac{\bar{x}_{kj} - \bar{x}_j}{m_k(\sigma_j + \sigma_0)}$$

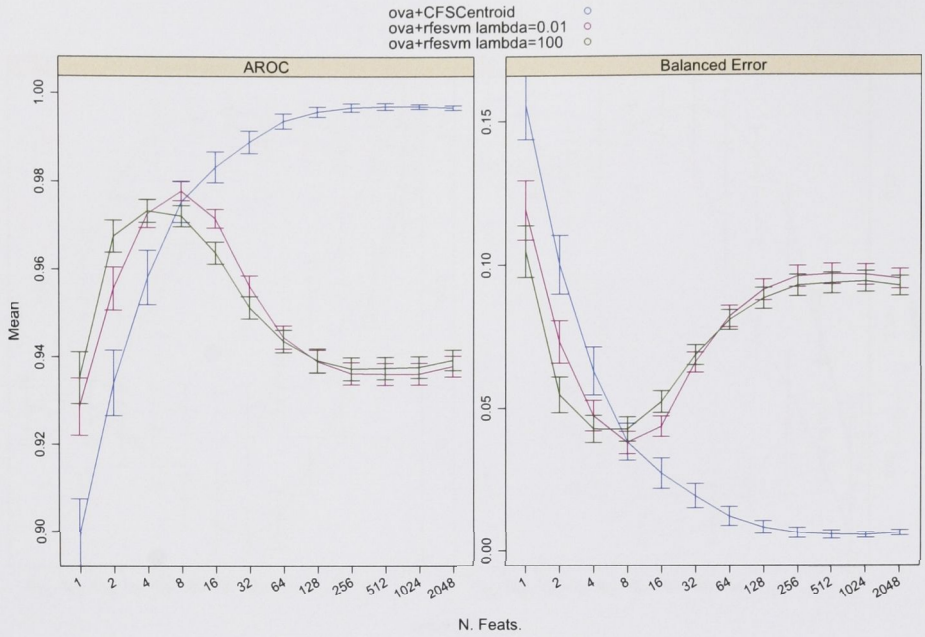


Figure 4.5: .632+ Bootstrap results for the centroid and RFE–SVM classifiers on the *lymphoma* dataset. Error bars are 95% confidence intervals.

where

$$\sigma = (\sigma_j)^m$$

$$\sigma_j = \frac{1}{n - |\mathbb{Y}|} \sum_{k \in \mathbb{Y}} \sum_{(\mathbf{x}, y=k) \in \mathcal{X}} (x_j - \bar{\mathbf{x}}_k)^2 \in \mathbb{R}$$

is the within-class standard deviation, $m_k = \sqrt{1/n + 1/n_k} \in \mathbb{R}$ is a class specific scaling factor, and $\sigma_0 \in R$ is a positive regularisation factor. Rearranging gives

$$\bar{\mathbf{x}}_k = \bar{\mathbf{x}} + m_k(\sigma + \sigma_0) \bullet \mathbf{d}.$$

The vector $\bar{\mathbf{x}}_k$ can be shrunk towards $\bar{\mathbf{x}}$ by reducing $\mathbf{d}$, i.e., by choosing

$$d'_{kj} = d_{kj} - \text{sign}(d_{kj}) \min(\Delta, |d_{kj}|)$$

for some $\Delta > 0$ the vector $\bar{\mathbf{x}}_k$ shrinks towards $\bar{\mathbf{x}}$. This produces the *shrunken class centroids*

$$\bar{\mathbf{x}}_k = \bar{\mathbf{x}} + m_k(\sigma + \sigma_0) \bullet \mathbf{d}'.$$

Features are eliminated when $d'_{kj} = 0$ for all k as feature j no-longer contributes

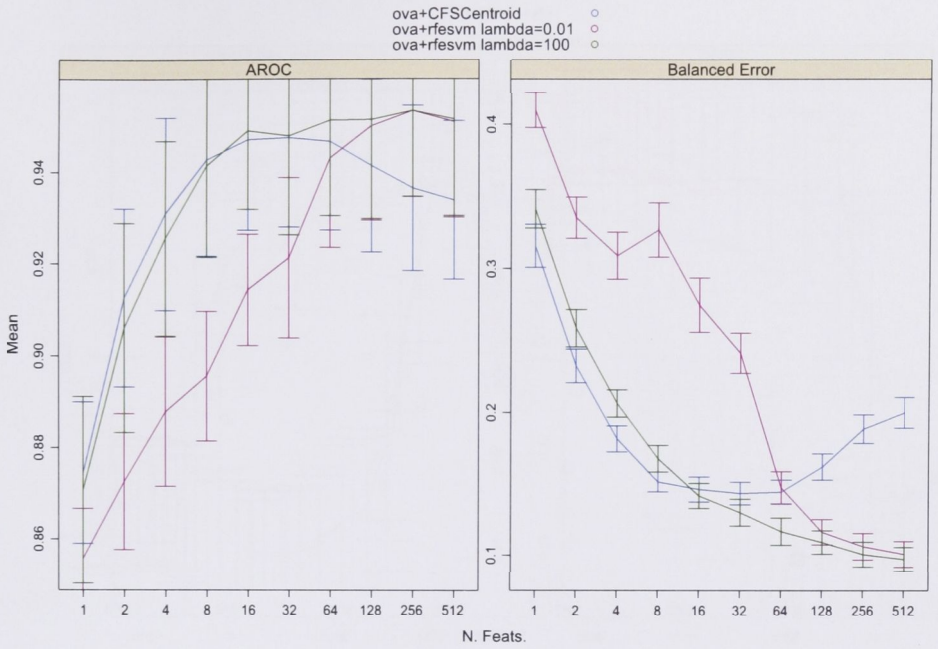


Figure 4.6: .632+ Bootstrap results for the centroid and RFE–SVM classifiers on the CUP dataset. Error bars are 95% confidence intervals.

to the discrimination between classes. Classification of a new sample is by the nearest shrunken centroid measured by *Mahalanobis distance*[4].

Figure 4.7 and Figure 4.8 show the same experiments as the previous section repeated with shrunken centroids (PAM). On these two-class datasets, PAM and the centroid method present here perform similarly. However this is not the case on multi-class datasets. Figure 4.9 and Figure 4.10 show the same comparison on the lymphoma and CUP multi-class datasets. An OVA was used only on the centroid method as PAM is a multi-class capable method. These results show the OVA centroid method performs significantly better than PAM after eliminating some features. Full bootstrap results are available in Appendix B.

4.8 Conclusions

The link between SVM and the centroid classifier was formally proved in the form of pointwise convergence, and in the convergence of discontinuous performance metrics when using analytic kernels. Using this link, an alternative to RFE–SVM

[4]The Mahalanobis distance between two vectors $\mathbf{x}, \mathbf{x}'$ is $d(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^* \Sigma^{-1} (\mathbf{x} - \mathbf{x}')}$ where Σ is the covariance matrix

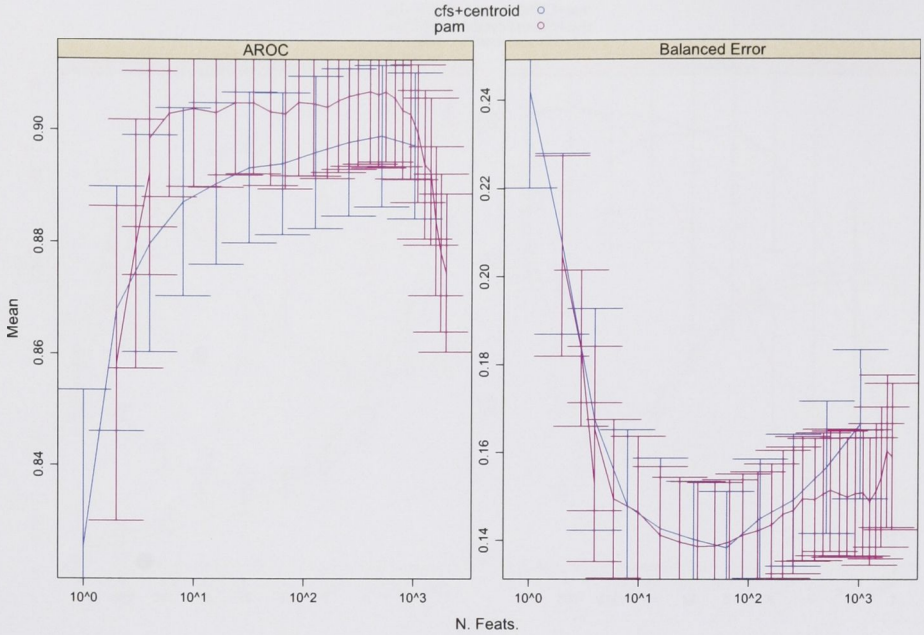


Figure 4.7: .632+ Bootstrap results for the centroid and PAM classifiers on the *colon* cancer dataset. Error bars are 95% confidence intervals.

was derived. This centroid based approach has proven to be very quick, and to perform well on the bioinformatics datasets evaluated. While the centroid approach will certainly not be the best choice for many problems, it does appear to be a good baseline performer for small-sample high-dimensional problems that are common in bioinformatics. One potential explanation is these datasets require a large amount of regularisation to bias model induction towards simple models to avoid overfitting problems. As the centroid is the high-regularisation limit of SVM, it would seem to be quite suitable for these types of problems. Finally, analytical equations were derived to evaluate the LTO and LOO error estimates for the centroid method. These equations decrease the computational cost of estimating generalisation error.

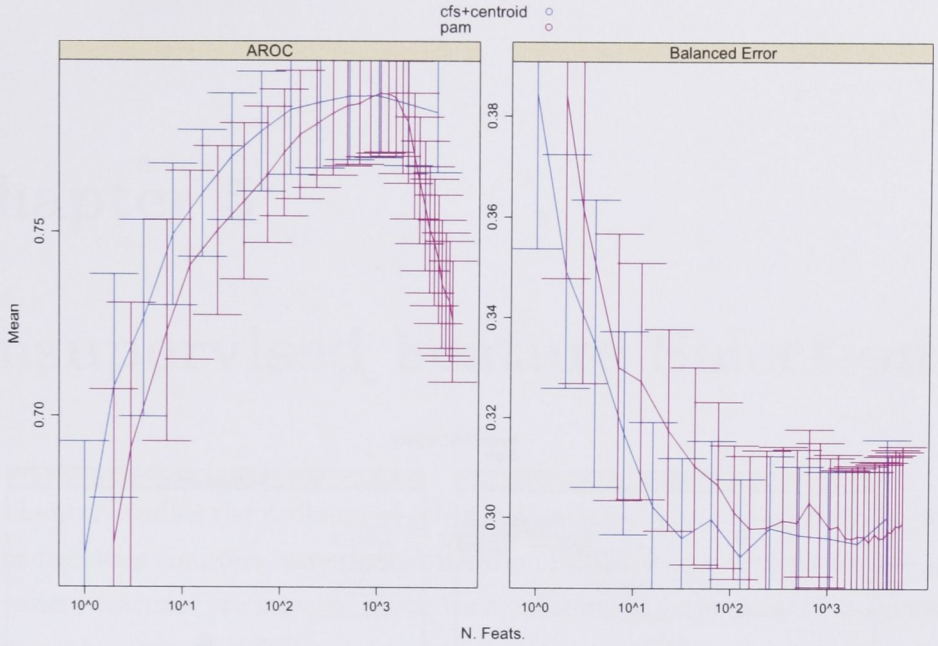


Figure 4.8: .632+ Bootstrap results for the centroid and PAM classifiers on the *van 't Veer* cancer dataset. Error bars are 95% confidence intervals.

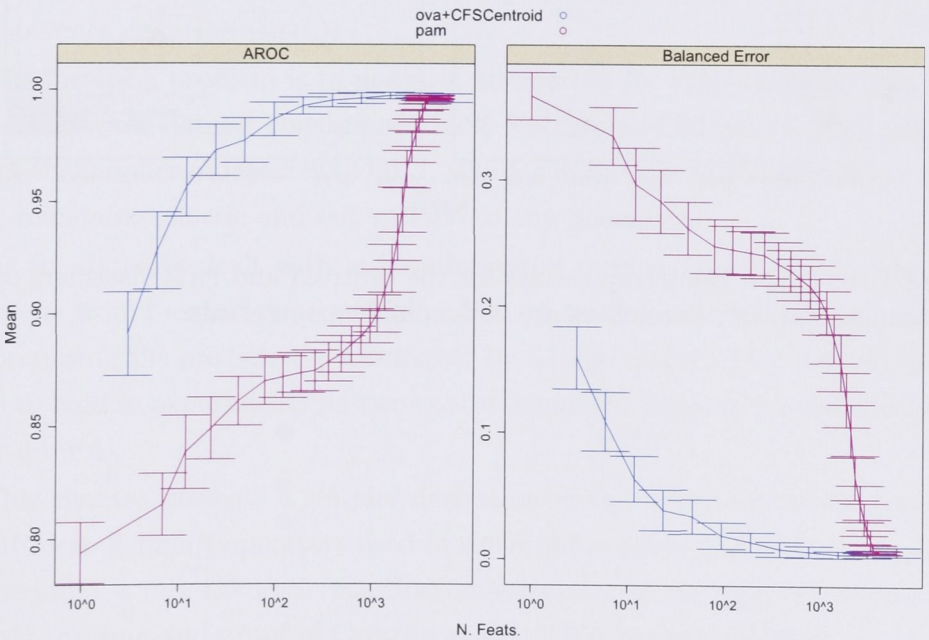


Figure 4.9: .632+ Bootstrap results for the centroid and PAM classifiers on the *lymphoma* cancer dataset. Error bars are 95% confidence intervals.

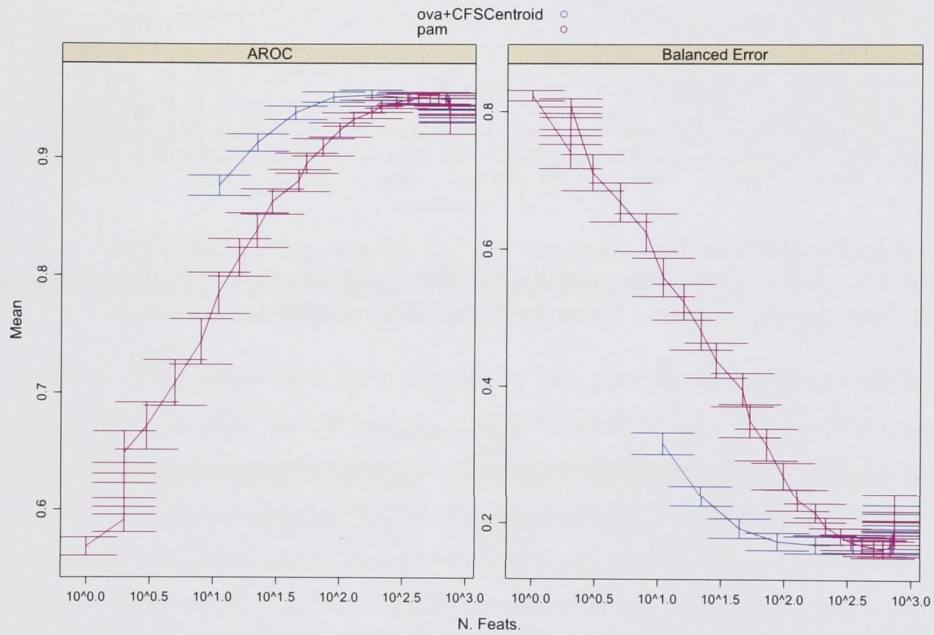


Figure 4.10: .632+ Bootstrap results for the centroid and PAM classifiers on the CUP cancer dataset. Error bars are 95% confidence intervals.

Chapter 5

Unsupervised Feature Selection

This chapter studies the problem of selecting features in an unsupervised context. All the previous chapters have dealt with the problem of supervised feature selection, where features are selected given their relation to some target (a continuous target in the case of QTL mapping and a categorical target in the case of the cancer classification datasets). In the case of unsupervised feature selection, features are selected without knowledge of a specific target by attempting to capture the maximum amount of information contained in the full dataset. This chapter approaches the unsupervised feature selection problem using the *Hilbert–Schmidt independence criterion* (HSIC).

The inspiring problem is to design a microarray for the sugarcane crop. The initial sugarcane dataset consists of 55,296 features and 80 plants. The aim is to reduce the number of features to 6972, which will fit on a single microarray plate, while remaining generic and not specific to any phenotypes.

As the datasets dealt with in bioinformatics tend to be extremely rank deficient, one would expect that a small subset of features can be selected without compromising the predictive performance by a large extent. This postulation appears to hold in experiments on two cancer genomics datasets previously studied in Chapter 4.

This chapter presents a simpler derivation of the HSIC that avoids the complex Hilbert–Schmidt operators used in the original paper (Gretton et al., 2005). Furthermore, a new theorem regarding an estimator of the HSIC is proven as the original theorem and proof of Gretton et al. (2005) has several flaws.

Using the HSIC estimator, an unsupervised feature selection algorithm is stated as a combinatorial optimisation problem. Using ideas from quantum physics, a *quantum annealing* optimiser is proposed for finding a good solution to the

problem. This optimisation problem is called the *unsupervised feature selection by the Hilbert–Schmidt independence criterion* (UBHSIC, pronounced ['u.bə-sik]). Several experiments on cancer and plant genomics datasets were conducted to evaluate the performance of selections obtained using UBHSIC.

5.1 The Hilbert–Schmidt Independence Criterion

The concept of independence in reproducing kernel Hilbert spaces will be introduced using *maximum mean discrepancy* (MMD). MMD is based on the following lemma that defines a sufficient and necessary conditions for the equivalence of any two probability measures.

Proposition 5.1. *Let P and Q be two Borel probability measures defined on the compact set $\mathbb{X} \subset \mathbb{R}^m$. Then $E_P[f] = E_Q[f]$ for all $f \in C^0(\mathbb{X}; \mathbb{R})$ iff $P = Q$, where C^0 is defined as in Definition 4.6.*

Proof. Let the assumptions of the theorem hold. First, note that the two expectations $E_P[f]$ and $E_Q[f]$ are both well defined[1] $\forall f \in C^0(\mathbb{X}; \mathbb{R})$ as $\mathbb{X}$ is compact and hence $f[\mathbb{X}]$ is bounded. The implication $P = Q \Rightarrow E_P[f] = E_Q[f]$ then follows immediately.

The reverse implication $E_P[f] = E_Q[f]$ $\forall f \in C^0(\mathbb{X}; \mathbb{R})$ is proved by Dudley (1987). □

The space of continuous functions $C^0(\mathbb{X}; \mathbb{R})$ is very rich and not convenient to work with directly. Instead, the dependence between probability measures is evaluated using functions from a *reproducing kernel Hilbert space* (RKHS, see Section 2.3).

Definition 5.2 (Universal kernels (Steinwart, 2001, 2005))**.** *Let $k\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$; $(\mathbf{x}, \mathbf{x}') \mapsto \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ be a kernel defined on a compact set $\mathbb{X}$ that generates the RKHS $\mathcal{H}$. The kernel k is called universal iff for all $f \in C^0(\mathbb{X}; \mathbb{R})$ and $\epsilon > 0$, there exists $g \in \mathcal{H}$ such that*

$$||f - g||_\infty < \epsilon,$$

where $C^0(\mathbb{X}; \mathbb{R})$ denotes the space of continuous maps from $\mathbb{X}$ to $\mathbb{R}$, and $||f||_\infty = \sup_{\mathbf{x}} |f(\mathbf{x})|$.

[1] An expectation of function $f\colon \mathbb{X} \to \mathbb{R}$ is called well defined if $E[f] \in \mathbb{R}$.

Universal kernels are particularly useful in this context because the function space is dense in the space of continuous maps. Using these kernels, the equivalence of two probability measures can be accurately determined by an extension of Proposition 5.1, as seen later. Two important universal kernels are defined as $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma||\mathbf{x} - \mathbf{x}'||^{\rho})$ where $\sigma > 0$ and $\rho \in \{1, 2\}$ (Steinwart, 2001, 2005). The choice of $\rho = 2$ produces the Gaussian RBF kernel (see Example 2.14), and $\rho = 1$ produces the Laplace kernel.

Using universal kernels, Proposition 5.1 can be extended. Let $\mathbb{X}$ be a compact set, $k\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$; $(\mathbf{x}, \mathbf{x}') \to \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ be a reproducing kernel with RKHS $\mathcal{H}$, and P be a Borel probability measure. Denote by ϕ_P the element of $\mathcal{H}$ such that

$$\langle \phi_P, f \rangle := E_{x \sim P}[\langle \phi(x), f \rangle] = E_P[f]. \tag{5.1}$$

The operator ϕ_P is guaranteed to exist by the Riesz representation theorem, provided $\sup_{||f||=1} |E_P[f]| < \infty$, which is guaranteed if $\phi\colon \mathbb{X} \to \mathcal{H}$ is continuous. The squared norm of ϕ_P follows as

$$\begin{aligned} ||\phi_P||^2 &= \langle \phi_P, \phi_P \rangle \\ &= E_{\mathbf{x} \sim P}[\langle \phi(\mathbf{x}), \phi_P \rangle] \\ &= E_{\mathbf{x} \sim P}[E_{\mathbf{x}' \sim P}[\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle]] \\ &= E_{\mathbf{x} \sim P}[E_{\mathbf{x}' \sim P}[k(\mathbf{x}, \mathbf{x}')] \\ &= E_{(\mathbf{x}, \mathbf{x}') \sim P \times P}[k(\mathbf{x}, \mathbf{x}')]. \end{aligned} \tag{5.2}$$

Furthermore, if Q is another Borel probablity measure, then

$$\begin{aligned} \langle \phi_P, \phi_Q \rangle &= E_{\mathbf{x} \sim P}[\langle \phi(x), \phi_Q \rangle] \\ &= E_{\mathbf{x} \sim P}[E_{\mathbf{x}' \sim Q}[\langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle]] \\ &= E_{(\mathbf{x}, \mathbf{x}') \sim P \times Q}[k(\mathbf{x}, \mathbf{x}')], \end{aligned} \tag{5.3}$$

where the last line follows by symmetry of k.

Lemma 5.3. *Let P and Q be two Borel probability measures defined on the compact set $\mathbb{X} \subset \mathbb{R}^m$, and $k\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$; $(\mathbf{x}, \mathbf{x}') \mapsto \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$ be a universal kernel with associated* RKHS *$\mathcal{H}$. Assume both ϕ_P and ϕ_Q are well defined. Then $\langle f, \phi_P - \phi_Q \rangle = 0$ for all $f \in \mathcal{H}$ iff $P = Q$.*

Proof. Let the assumptions of the theorem hold. The implication $P = Q \Rightarrow$

$\langle f, \phi_P - \phi_Q \rangle = 0 \; \forall f \in \mathcal{H}$ is clear as $\phi_P - \phi_Q$ is well defined by the assumption.

For the reverse implication, assume $\langle f, \phi_P - \phi_Q \rangle = 0$ but $P \neq Q$. The proof is by showing a contradiction follows. It follows from the reproducing kernel property that

$$\begin{aligned} 0 &= \langle f, \phi_P - \phi_Q \rangle && (5.4) \\ &= \langle f, \phi_P \rangle - \langle f, \phi_Q \rangle \\ &= E_{\mathbf{x} \sim P}[\langle f, \phi(\mathbf{x}) \rangle] - E_{\mathbf{x} \sim Q}[\langle f, \phi(\mathbf{x}) \rangle] \\ &= E_P[f] - E_Q[f] && (5.5) \end{aligned}$$

for any $f \in \mathcal{H}$. However, since $P \neq Q$, from Proposition 5.1 there exists a continuous function $g \in C^0(\mathbb{X}; \mathbb{R})$ such that

$$E_P[g] \neq E_Q[g]. \tag{5.6}$$

By the definition of universality (Definition 5.2), for any $\epsilon > 0$ there exists $g' \in \mathcal{H}$ such that $||g - g'||_\infty < \epsilon$, hence

$$\begin{aligned} |E_P[g] - E_Q[g]| &\leq |E_P[g'] - E_Q[g']| + (E_P[1] + E_Q[1])\epsilon \\ &= |E_P[g'] - E_Q[g']| + 2\epsilon \\ &= 2\epsilon \end{aligned}$$

from Equation 5.5. This inequality contradicts (5.6) for any $\epsilon > 0$. □

Given this lemma, two probability distributions can be compared by

$$\sup_{f \in \mathcal{H}} \langle f, \phi_P - \phi_Q \rangle .$$

This quantity is zero iff the distributions are equivalent, provided $\mathcal{H}$ admits a universal kernel. This quantity is called the *maximum mean discrepancy* (MMD, Gretton et al. (2006)).

Let PQ be the joint probability measure with marginals P and Q defined on $\mathcal{X}$. A dependence measure between PQ and the product of the marginals P and Q can be easily defined using the MMD. This is similar to the *mutual information* measure in information theory, which is the KL-divergence (see Definition 2.25)

between the joint distribution and the product of the marginals. Let

$$k'' \colon (\mathbb{X} \times \mathbb{X}) \times (\mathbb{X} \times \mathbb{X}) \to \mathbb{R};\ ((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) \mapsto \langle \phi(\mathbf{x}, \mathbf{y}), \phi(\mathbf{x}', \mathbf{y}') \rangle$$

be a reproducing kernel with RKHS $\mathcal{H}$. Assume that ϕ_{PQ} and $\phi_{P \times Q}$ exist. Define

$$D_{\text{dep}} := ||\phi_{PQ} - \phi_{P \times Q}||$$

as a measure of dependence between P and Q. Taking the square of this quantity gives an expression in terms of the kernel:

$$\begin{aligned} {D_{\text{dep}}}^2 &= ||\phi_{PQ} - \phi_{P \times Q}||^2 \\ &= ||\phi_{PQ}||^2 - 2 \langle \phi_{PQ}, \phi_{P \times Q} \rangle + ||\phi_{P \times Q}||^2 \\ &= E_{PQ \times PQ}[k''] - 2E_{PQ \times (P \times Q)}[k''] + E_{(P \times Q) \times (P \times Q)}[k''], \end{aligned}$$

where the last statement follows from the application of Equation 5.2 and Equation 5.3.

Proposition 5.4. *Let P and Q be two Borel probability measures defined on the compact set $\mathbb{X} \subset \mathbb{R}^m$, PQ be the joint probability measure, and $k \colon \mathbb{X} \to \mathcal{H}$ be a universal kernel. Assume the existence of ϕ_{PQ} and $\phi_{P \times Q}$. Then, $D_{dep} = 0$ iff P and Q are independent.*

Proof. The two probabilities P and Q are independent iff $PQ = P \times Q$. Expanding the definition of D_{dep} gives

$$\begin{aligned} D_{\text{dep}} &= ||\phi_{PQ} - \phi_{P \times Q}|| \\ &= \sup_{f \in \mathcal{H}; ||f|| \leq 1} \langle f, \phi_{PQ} - \phi_{P \times Q} \rangle . \end{aligned} \tag{5.7}$$

The RHS of Equation 5.7 is zero iff $PQ = P \times Q$ by Lemma 5.3. □

If k'' is chosen such that

$$k''((\mathbf{x}, \mathbf{y}), (\mathbf{x}', \mathbf{y}')) = k(\mathbf{x}, \mathbf{x}') l(\mathbf{y}, \mathbf{y}'),$$

then the squared dependence becomes

$${D_{\text{dep}}}^2 = E_{PQ \times PQ}[kl] - 2E_{PQ \times (P \times Q)}[kl] + E_{(P \times Q) \times (P \times Q)}[kl] \tag{5.8}$$

dependent on P, Q, k, and l. Note that k'' is a reproducing kernel if k and l are reproducing kernels by the Moore–Aronszajn theorem as the product kl is positive semi-definite (Cristianini and Shawe-Taylor, 2000). This quantity is equivalent to the *Hilbert–Schmidt independence criterion* (HSIC, Gretton et al. (2005)). Unfortunately, it is difficult to calculate explicitly as P and Q are unknown, hence an estimator is required. The next theorem provides a biased estimator for this quantity.

Theorem 5.5. *Let P and Q be two Borel probability measures defined on compact set $\mathbb{X}$, $\mathcal{X} := \{\mathbf{x}_i\}_{i=1}^n \sim P$, and $\mathcal{Y} := \{\mathbf{y}_i\}_{i=1}^n \sim Q$. Choose continuous reproducing kernels $k\colon \mathbb{X} \to \mathcal{H}$ and $l\colon \mathbb{X} \to \mathcal{H}'$. Denote the kernel matrices as $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ and $L_{ij} := l(\mathbf{y}_i, \mathbf{y}_j)$. Then*

$$D^2_{dep} = E_{PQ^n}[\frac{1}{(n)_2} \operatorname{tr}(KHLH)] + O(\frac{1}{n})$$

where $H = Id - \frac{1}{n}\mathbf{1}\mathbf{1}^*$, $(n)_r = \frac{n!}{(n-r)!}$ is the Pochhammer symbol, and tr() *denotes the matrix trace (i.e., the sum of the diagonal).*

A similar theorem was proposed by Gretton et al. (2005, Theorem 1), however the proof presented in section A.2 on page 75 was flawed; the last term in their expansion of $\operatorname{tr}(KHLH)$ is incorrect (compare Equation 5.9 with their expansion), and a different scaling factor for each term was needed. Here a slightly different theorem with a rigourous proof is presented.

Proof. Let $\mathbf{i}_r^n$ be the set of all r-tuples drawn from $\{1, \ldots, n\}$ without replacement, which has cardinality $(n)_r$. Note that

$$(n)_r = \prod_{i=0}^{r-1}(n-i) = O(n^r),$$

and

$$\frac{(n)_{r+1}}{(n)_r} = \frac{n!}{(n-r-1)!}\frac{(n-r)!}{n!} = (n-r).$$

By definition of H, the following expression is obtained:

$$\begin{aligned}\frac{1}{(n)_2}\operatorname{tr}(KHLH) &= \frac{1}{(n)_2}\operatorname{tr}(KL) \\ &\quad - \frac{1}{n(n)_2}\operatorname{tr}(KL\mathbf{1}\mathbf{1}^*) - \frac{1}{n(n)_2}\operatorname{tr}(K\mathbf{1}\mathbf{1}^*L) \\ &\quad + \frac{1}{n^2(n)_2}\operatorname{tr}(K\mathbf{1}\mathbf{1}^*L\mathbf{1}\mathbf{1}^*) \\ &= \frac{1}{(n)_2}\operatorname{tr}(KL) - \frac{2}{n(n)_2}\mathbf{1}^*KL\mathbf{1} + \frac{1}{n^2(n)_2}\mathbf{1}^*K\mathbf{1}\mathbf{1}^*L\mathbf{1}. \qquad (5.9)\end{aligned}$$

The proof is by considering the expectation of each term in Equation 5.9. Consider the expectation of the first term:

$$\begin{aligned}T_1 := \frac{1}{(n)_2}E_{PQ^n}\operatorname{tr}(KL) &= \frac{1}{(n)_2}E_{PQ^n}[\sum_{i,j}K_{ij}L_{ji}] \\ &= \frac{1}{(n)_2}E_{PQ^n}[\sum_{(i,j)\in\mathbf{i}_2^n}K_{ij}L_{ji} + \sum_i K_{ii}L_{ii}] \\ &= \frac{1}{(n)_2}E_{PQ^n}[\sum_{(i,j)\in\mathbf{i}_2^n}K_{ij}L_{ji} + O(n)]\end{aligned}$$

This equality holds as any the sum over any two indices i, j can be expressed as the sum over 2-tuples contained in $\mathbf{i}_r^n$ plus a correction term of $\sum_i K_{ii}L_{ii}$. In this particular case, the correction term is of $O(n)$ as the kernel functions have universally bounded magnitudes (being continuous on compact domain $\mathbb{X}\times\mathbb{X}$). As $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $L_{ij} = l(\mathbf{y}_i, \mathbf{y}_j)$, and samples $(\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}_j, \mathbf{y}_j) \sim PQ \times PQ$ are IID, the expectation of each term under the sum is identical. Hence

$$\begin{aligned}T_1 &= \frac{1}{(n)_2}(n)_2E_{PQ\times PQ}[kl] + O(\frac{1}{n}) \\ &= E_{PQ\times PQ}[kl] + O(\frac{1}{n}).\end{aligned}$$

This is the first term of Equation 5.8. □

Now, consider the expectation of the second term in Equation 5.9:

$$T_2 := \frac{2}{n(n)_2} E_{PQ^n}[\mathbf{1}^* K L \mathbf{1}] = \frac{2}{n(n)_2} E_{PQ^n}[\sum_{i,j,a} K_{ij} L_{ja}]$$
$$= \frac{2}{n(n)_2} E_{PQ^n}[\sum_{(i,j,a)\in \mathbf{i}_3^n} K_{ij} L_{ja} + O(n^2)],$$

where the last line follows similarly to the first case (as the correction term is a sum over n pairs and n samples, it is of order $O(n^2)$). Continuing gives

$$\begin{aligned} T_2 &= \frac{2}{n(n)_2}(n)_3 E_{(\mathbf{x},\mathbf{y})\sim PQ}[E_{\mathbf{x}'\sim P}[k(\mathbf{x}', \mathbf{x})] E_{\mathbf{y}'\sim Q}[l(\mathbf{y}, \mathbf{y}')]] + O(\frac{1}{n}) \\ &= \frac{2(n-2)}{n} E_{PQ\times(P\times Q)}[kl] + O(\frac{1}{n}) \\ &= 2E_{PQ\times(P\times Q)}[kl] - \frac{4}{n} E_{PQ\times(P\times Q)}[kl] + O(\frac{1}{n}) \\ &= 2E_{PQ\times(P\times Q)}[kl] + O(\frac{1}{n}). \end{aligned}$$

□

Consider the expectation of the last term in Equation 5.9:

$$T_3 := \frac{1}{n^2(n)_2} E_{PQ^n}[\mathbf{1}^* K \mathbf{1}\mathbf{1}^* L \mathbf{1}] = \frac{1}{n^2(n)_2} E_{PQ^n}[\sum_{i,j,a,b} K_{ij} L_{ab}]$$
$$= \frac{1}{n^2(n)_2} E_{PQ^n}[\sum_{(i,j,a,b)\in \mathbf{i}_4^n} K_{ij} L_{ab} + O(n^3)],$$

which follows similarly to the previous cases,

$$\begin{aligned} T_3 &= \frac{(n)_4}{n^2(n)_2} E_{(P\times Q)\times(P\times Q)}[kl] + O(\frac{1}{n}) \\ &= \frac{(n-2)(n-3)}{n^2} E_{(P\times Q)\times(P\times Q)}[kl] + O(\frac{1}{n}) \\ &= E_{(P\times Q)\times(P\times Q)}[kl] + O(\frac{1}{n}). \end{aligned}$$

□

The consequence of this theorem is that dependence can be maximised by maximising the quantity $\mathrm{tr}(KHLH)$. Recall from Proposition 5.4 that D_{dep} is

zero iff there is no dependence provided the kernel is a universal kernel. Any kernel can be used with the consequence that D_{dep} loses the uniqueness property, i.e., if P and Q are independent then $D_{\text{dep}} = 0$ but the reverse implication does not hold. If the kernel is not universal and D_{dep} is zero for two probabilities P and Q, then P and Q are said to be equivalent *with respect to the function class* admitting the kernel.

5.2 Quantum Annealing

Using the HSIC, the compression task can now be specified as the following optimisation problem. This method is called *unsupervised feature selection by the Hilbert–Schmidt independence criterion* (UBHSIC, pronounced ['u.bə-sik]).

Definition 5.6 (UBHSIC optimisation problem). *Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{X}$ be a dataset and $k\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a kernel. The* UBHSIC *selection is given by the solution to the optimisation problem*

$$\max_{\theta} \operatorname{tr}(K^{\theta} H K H)$$

$$\text{such that}$$

$$|\theta| = m',$$

where $K^{\theta}_{ij} = k((x_{if})_{f\in\theta}, (x_{jf})_{f\in\theta})$ is the kernel matrix restricted to the features in set $\theta \neq \emptyset$, $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel matrix over the full data set, and $0 < m' < m$ is the number of features desired.

This is a combinatorial optimisation problem and thus has no analytical solution. A good solution can be found by simply applying RFE whereby a few features are found by commencing with all features and iteratively removing the feature which leads to the highest HSIC (see Section 2.5.2). Forward selection could also be used, which is similar to RFE but commences with no features and iteratively adds the feature that increases the HSIC the most. In our applications these strategies are computationally unfeasible as RFE and forward selection would require $(m)_{m-m'}$ and $(m)_{m'}$ iterations.

5.2.1 Diffusion Monte Carlo

To solve this combinatorial optimisation problem, some ideas from quantum physics will be used. In particular, the *diffusion Monte Carlo* (DMC) method will be

derived via the Feynman path-integral, following Kosztin et al. (1997). Consider a single particle with mass m moving in one dimension with position[2] x. The time-dependent Schrödinger equation defines the wavefunction $\psi(x,t)$ of the particle as

$$i\hbar\frac{\partial\psi}{\partial t} = \hat{H}\psi,$$

where t is time and the Hamiltonian $\hat{H}$ is

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x),$$

where $\hbar$ is the reduced Plank constant[3]. The first term of the Hamiltonian is the *kinetic energy* and the second term $V(x)$ is the *potential energy*. The formal solution given an initial state $\psi(x,0)$ is

$$\psi(x,t) = e^{-\frac{i}{\hbar}\hat{H}t}\psi(x,0). \tag{5.10}$$

If the particle is constrained to a finite spatial domain, then the potential $V(x)$ as $x \to \pm\infty$ diverges to infinity and the state $\psi(x,0)$ can be decomposed into *eigenstates* ϕ_i such that

$$\psi(x,0) = \sum_{i=0}^{\infty} c_i\phi_i,$$

where ϕ_i are the solutions of the *time-independent* Schrödinger equation

$$\hat{H}\phi_i = \epsilon_i\phi_i.$$

Expanding Equation 5.10 using the eigenstates yields

$$\psi(x,t) = \sum_{i=0}^{\infty} c_i e^{-\frac{i}{\hbar}\epsilon_i t}\phi_i,$$

where ϕ_i are eigenvectors and ϵ_i are eigenvalues.

The *imaginary-time* Schrödinger equation is obtained by introducing a *Wick rotation of time* by defining $t = i\tau$, and coupled with a shift in the energy scale

[2]there is a clash of notation here; throughout this section x and $\mathbf{x}$ refer to particle positions not samples to remain consistent with standard quantum physics notation

[3]The actual value of the constant is not important for the final optimiser.

$V(x) \to V(x) - E_r$ gives

$$\hbar \frac{\partial \psi}{\partial \tau} = \frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2} - [V(x) - E_r]\psi \tag{5.11}$$

with a wavefunction expansion of

$$\psi(x, \tau) = \sum_{i=0}^{\infty} c_i e^{-\frac{\epsilon_i - E_r}{\hbar}\tau} \phi_i.$$

Let the labelling of the eigenstates be in the order of increasing energy, i.e., where

$$\epsilon_0 < \epsilon_1 \leq \epsilon_2 \leq \dots.$$

The *ground state energy* is then ϵ_0, which we seek. Furthermore, several observations regarding the asymptotic behaviour for $\tau \to \infty$ can be made:

1. if $E_r > \epsilon_0$ then $\psi(x, \tau) \to \infty$ as $\tau \to \infty$;

2. if $E_r < \epsilon_0$ then $\psi(x, \tau) \to 0$ as $\tau \to \infty$;

3. if $E_r = \epsilon_0$ then $\psi(x, \tau) = c_0 \phi_0(x)$ as $e^{-\frac{\epsilon_i - E_r}{\hbar}\tau} \to 0$ as $\tau \to \infty$ for all $i > 0$.

This asymptotic behaviour is the core idea of the DMC method and shows that ψ converges to the ground state wavefunction ϕ_0 regardless of the initial wavefunction $\psi(x, 0)$ in the long time limit if $E_r = \epsilon_0$, provided c_0 is sufficiently large. Indeed, Schrödinger's equation in this case can be considered a diffusion equation, with the wavefunction ψ modelling the density of particles. In light of this viewpoint, it is clear that the density of particles increases for $E_r > \epsilon_0$ and decreases for $E_r < \epsilon_0$.

One therefore seeks to find $\psi(x)$ by integrating the imaginary time Schrödinger equation (Equation 5.11) for an arbitrary reference energy E_r given an initial wavefunction $\psi(x, 0)$. This can be done by *time slicing*, which gives rise to the Feynman path integral. Using the path integral the wavefunction can be written as

$$\begin{aligned}\psi(x, \tau) = \lim_{N \to \infty} \int_{\infty}^{\infty} dx_0 \int_{-\infty}^{\infty} dx_1 \dots \int_{-\infty}^{\infty} dx_{N-1} \left(\frac{m}{h\Delta\tau}\right)^{\frac{N}{2}} \\ \times \exp\left(-\frac{\Delta\tau}{\hbar} \sum_{n=1}^{N} \left[\frac{m}{2\Delta\tau^2}(x_n - x_{n-1})^2 + V(x) - E_r\right]\right) \psi(x_0, 0)\end{aligned}$$

where $\Delta\tau = \frac{\tau}{N}$ is a small time step. Defining

$$P(x_n, x_{n-1}) = \left(\frac{m}{\hbar\Delta\tau}\right)^{\frac{1}{2}} \exp\left(-\frac{m(x_n - x_{n-1})^2}{2\hbar\Delta\tau}\right)$$

and

$$W(x_n) = \exp\left(-\frac{V(x_n) - E_r}{\hbar}\Delta\tau\right)$$

simplifies the wave equation to

$$\psi(x, \tau) = \lim_{N\to\infty} \int_{-\infty}^{\infty} \left(\prod_{n=0}^{N-1} dx_n\right) \prod_{n=1}^{N} W(x_n) P(x_n, x_{n-1}) \psi(x_0, 0). \tag{5.12}$$

It is clear the function $P(x_n, x_{n-1})$ is a normally distributed transition probability function describing the probability of transitioning from position x_{n-1} to x_n. The $W(x_n)$ function is a *weight function* that depends on the potential energy $V(x)$ and the arbitrary reference value E_r. The weight function $W(x_n)$ is an unscaled member of the exponential family, i.e., where the log partition function is unknown.

The integral in Equation 5.12 cannot be evaluated analytically for most cases and an approximation must be made. This is done through *Monte Carlo* (MC) sampling, yielding the equation

$$\psi(x, \tau) \approx \frac{1}{M} \sum_{i=1}^{M} \prod_{n=1}^{N} W(x_n) \psi(x_0, 0)$$

where each $(x_1, \ldots, x_n)$ is drawn from the probability distribution $\prod_{n=1}^{N} P(x_n, x_{n-1})$.

This MC equation allows us to calculate $\psi(x, \tau)$ for any time τ, however it does not provide the ground state energy ϵ_0 and its wavefunction for $\tau \to \infty$. To solve these problems simultaneously consider the wavefunction ψ as a probability density. Thus, the path integral equation becomes a product of a series of sequential stochastic processes, which can be modelled numerically.

The initial wavefunction $\psi(x_0, 0)$ defines the starting position for a collection of particles and may be chosen as the Dirac δ-function, which places all the particles at the same initial position, or it may be chosen such that the initial particles are randomly positioned across the energy landscape. Each MC step now consists of a series of *diffusive displacements*, where each particle diffuses to a new location according to the probability density $P(x_n, x_{n-1})$. Finally, instead

of accumulating the weights $W(x_n)$ of each particle, particles either *branch* into 2 or more new particles or are *removed* with probability proportional to the weights. Thus, particles positioned in low energy regions are likely to branch while particles in higher energy regions are likely to disappear. The reference energy E_r balances the ratio of branching to removing, and is updated at each iteration to the mean potential energy of each particle. This process of diffusion, branching, and removing based on the reference energy is the *diffusion Monte Carlo* (DMC) method (Kosztin et al., 1997).

5.2.2 The UBHSIC Optimiser

To use DMC to optimise our combinatorial optimisation function, the *kinetic energy potential energy* terms of the Hamiltonian and a state representation for the feature subset must be defined. As features can be either selected or not selected, the state can be modelled as an *Ising spin model* where the i^{th} particle is represented the state representation $\mathbf{S}_i := (S_{ij})^m \in \{1, -1\}^m$, where $S_{ij} = +1$ or $S_{ij} = -1$ if the j^{th} feature is selected or unselected, with the constraint $|\{j|s_{ij} = 1\}| = m'$. The constraint can be satisfied by only proposing moves that do not break the constraint, i.e., each move consists of flipping a 1 bit to -1, and a -1 bit to 1. The diffusion of a particle is now easily defined using the Boltzmann distribution as an acceptance probability; given a proposed diffusive move of particle i from $\mathbf{S}_i$ to $\mathbf{S}'_i$, the move is accepted with probability 1 if $V(\mathbf{S}'_i) < V(\mathbf{S}_i)$, and with

$$\exp(-\frac{V(\mathbf{S}'_i) - V(\mathbf{S}_i)}{\gamma})$$

otherwise. The parameter γ is the "temperature" and is initially set high to allow for large diffusions. As the optimisation progresses, it is *annealed* down to narrow the size of the neighbourhood search according to a logarithmic "cooling" schedule $\gamma = \frac{2000}{\log(10\lfloor(t-1)/10\rfloor+e)}$, where $t \in \{1, 2, \ldots\}$ is current iteration. Due to the annealing behaviour, this is called a *quantum annealing* (QA) optimiser.

Given a state representation, the potential energy follows as

$$V(\mathbf{S}_i) = -\operatorname{tr}(K^{\theta_i} HKH),$$

where $\theta_i := \{j|S_{ij} = 1\}$. The negative arises as the optimiser *minimises* energy whereas the *maximum* dependence is sought.

The penultimate step is to define an initial wavefunction for initialisation of

the particles. As mentioned previously, the wavefunction can be chosen as the Dirac δ-function, however care must be taken to centre the δ-function at a good starting position. Consider now UBHSIC for a linear kernel $K = XX^*$:

$$\begin{aligned} \mathrm{tr}(KHKH) &= \mathrm{tr}(XX^*M) \\ &= \mathrm{tr}(X^*MX) \\ &= \sum_j \mathbf{x}^{(j)*} M \mathbf{x}^{(j)} \end{aligned}$$

where $M = HKH$ and $\mathbf{x}^{(j)}$ denotes the j^{th} column vector of X. Thus, in the case of a linear kernel the features are independent and can be ranked by $\mathbf{x}^{(j)*} M \mathbf{x}^{(j)}$ and selected in a greedy fashion. Indeed, in this case the QA algorithm is unnecessary as this process achieves the global minimum. However, for non-linear kernels it provides a good initial position for the particles – the matrix M can be calculated using the non-linear kernel of choice, and the features ranked according to $\mathbf{x}^{(j)*} M \mathbf{x}^{(j)}$ and selected greedily to produce the initial position.

Finally, consider the weight function which defines the branching and removal behaviour of the particles. To simplify matters considerably, the branch/remove procedure will be simplified to the following:

1. if $V(\mathbf{x}_i) \leq E_r$, the particle branches into 2 particles that then undergo diffusion;

2. if $V(\mathbf{x}_i) > E_r$ the particle is removed.

E_r is chosen as the average energy for the current set of particles, with restrictions to keep the number of particles within an acceptable range as the number of particles can increase exponentially when E_r is close to the ground state energy. Algorithm 5.1 gives an explicit description of the complete algorithm.

This optimisation method is related to the family of "Go With the Winners" algorithms (Aldous and Vazirani, 1994) and genetic algorithms, however without the fixed population size as the number of particles is free to change.

5.3 Results and Discussion

UBHSIC was evaluated on several cancer genomics datasets and a sugarcane dataset from DArT. Several kernels were experimented with to determine the effects of different kernels. These kernels are defined as follows:

RBF: $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma||\mathbf{x}-\mathbf{x}'||_2^2)$ with σ set as the inverse median of the squared distances $||\mathbf{x} - \mathbf{x}'||_2^2$ between pairs in the dataset

Linear: $k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$

Polynomial: $k(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^d$ for $d \in \{2, 3\}$

Variance: $k(\mathbf{x}, \mathbf{x}') = \frac{\langle \mathbf{x}, \mathbf{x}' \rangle^2}{\langle \mathbf{x}, \mathbf{x} \rangle \langle \mathbf{x}', \mathbf{x}' \rangle}$

The last kernel, the variance kernel, was selected to favour decorrelated selections. This is important for the sugarcane dataset where the subset is required to be highly decorrelated for elimination of near identical probes. The preference towards decorrelation, indirectly encoded as $\frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\sqrt{\langle \mathbf{x}, \mathbf{x} \rangle \langle \mathbf{x}', \mathbf{x}' \rangle}}$, is the cosine of the angle between the two vectors $\mathbf{x}$ and $\mathbf{x}'$. As adding a feature highly correlated with another already selected feature will not affect the angle between the vectors as much as an orthogonal feature, one may postulate that the kernel used with UBHSIC will produce highly decorrelated selections.

The parameters for the QA algorithm were set as follows. A convergence criteria of 100 iterations with no improvement was specified and the number of particles constrained between 4 and 200 particles. The maximum number of iterations was set to 10,000 iterations, though this was never reached as the convergence criteria was always met first. See Algorithm 5.1 for a pseudo-code description of the algorithm.

5.3.1 Cancer Genomics

Two cancer genomics datasets previously studied in Chapter 4 – the van 't Veer dataset and the Colon dataset – and a microarray cancer of unknown primary (CUP) dataset (Tothill et al., 2005) that precedes the QPCR–CUP dataset studied in Chapter 4 were used to evaluate UBHSIC. For a description of the van 't Veer and Colon cancer datasets, see Section 4.7.

The CUP dataset is a multiclass classification dataset where the aim is to develop a predictor for the site of origin of a tumour from a microarray of a sample. It consists of 14 classes, 220 samples, and 9630 features. Each class is not represented equally, with the smallest class containing only 3 samples and the largest containing 34. The main difference with the QPCR–CUP dataset studied in Chapter 4 is it has considerably more features (9630 instead of 740).

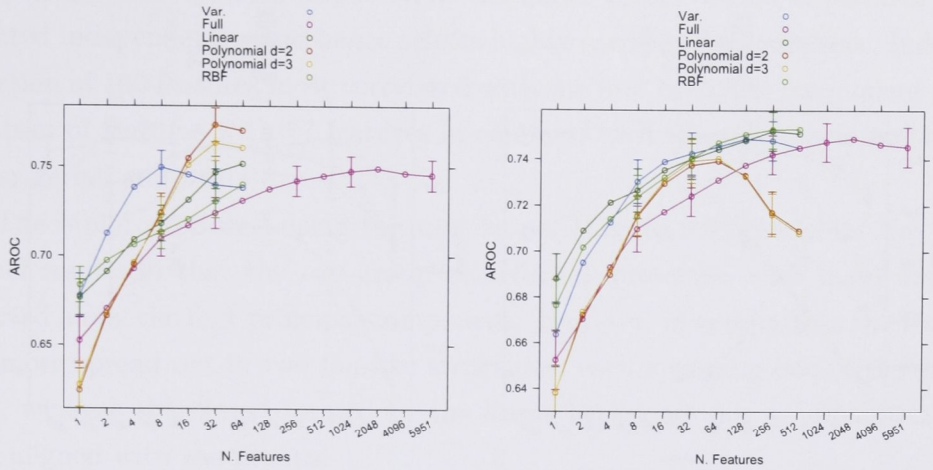
The centroid classifier and feature selector (see Chapter 4) was used to analyse the full dataset and the reduced datasets obtained using UBHSIC and the various kernels. For the multiclass CUP dataset, a one-vs-all (OVA, see Section 2.8) architecture was used in conjunction with the centroid method to produce predictions. The ϵ-0 bootstrap estimator with 200 repetitions was used to evaluate the generalisation performance.

Each dataset was analysed by applying UBHSIC with the various kernels to reduce the full dataset, and then comparing the classification performance before and after reduction. Supervised filtering was used in both cases to determine the effect of further post-filtering after applying UBHSIC, and to compare the performance against a fully supervised approach.

Figure 5.1 shows the results of pre-filtering using UBHSIC down to 50 (Subfigure 5.1a) and 500 features (Subfigure 5.1b) followed by supervised filtering and classification using the centroid method. With the reduction to 500 features, the linear, RBF and variance kernels do very well; they achieve a level of performance equivalent to the full dataset, and exceed the full dataset performance at lower numbers of features. The two polynomial kernels initially do not perform well, but after feature selection the performance equals that of the other kernels and the full dataset. Under aggressive reduction down to 50 features, somewhat surprising results are obtained; the maximum performance achieved is *better* than the full dataset at the 95% statistical significance level using the 2^{nd} degree polynomial kernel, despite the using only 32 features. Furthermore, the variance kernel achieves the best performance at the 8 features operating point. Both are significantly less than the original 70 genes proposed for classification by the original paper (van 't Veer et al., 2002).

Performing the same experiments on the colon cancer dataset yielded the results in Figure 5.2. Again, strong performance when using the variance and RBF kernels is observable in Subfigure 5.2b; RBF produced very good results after further supervised filtering down to a few features (4) while the variance kernel produced very similar results to the full dataset. The linear and polynomial kernels do not perform well on this dataset; this is supported by the results shown in Subfigure 5.2a where the linear and polynomial kernels again perform poorly, but the RBF and variance kernels perform well.

Finally, the results of applying the unsupervised feature selection to the CUP dataset is shown in Figure 5.3. As this dataset is a larger dataset (220 samples) than both the colon and van 't Veer datasets, a less aggressive filtering was



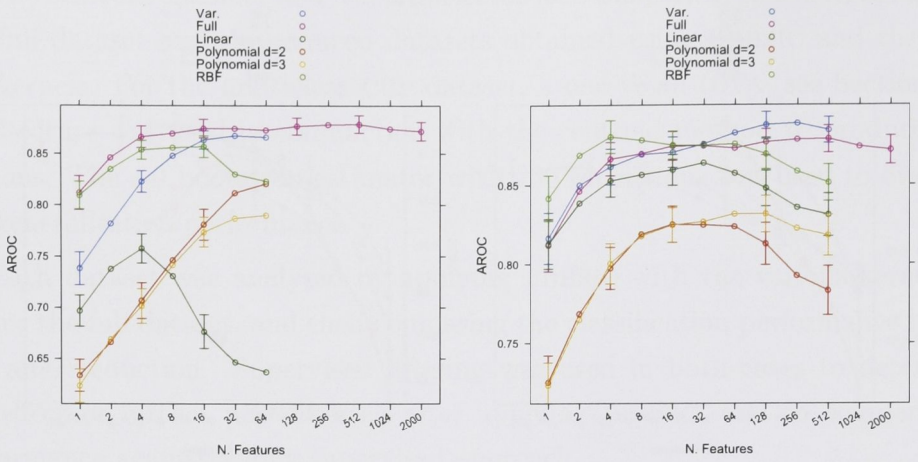
(a) Aggressive reduction to 50 features (b) Reduction to 500 features

Figure 5.1: van 't Veer dataset with centroid classifier and feature selector. Results are using the ϵ-0 bootstrap with 200 repetitions. Error bars show 95% confidence interval. Subfigure (a) shows the performance of the dataset reduced to 50 features using the UBHSIC procedure and various kernels. Each plot corresponds to a different kernel, with the purple plot corresponding to the CFSCentroid method on the entire dataset (i.e., without prefiltering using UBHSIC). The 5 plots where prefiltering using UBHSIC was used do not extend above 50 features, and further supervised filtering using the CFS was applied to determine the maximum performance achievable from the reduced datasets. Subfigure (b) is similar to subfigure a, except with less aggressive UBHSIC reduction (reduced to 500 features instead of 50).

applied. Subfigure 5.3c shows the performance curves obtained after filtering to 1000 features. At 1000 features, the variance kernel produced a subset with equivalent performance to the full dataset. After aggressive reduction to 100 features, the performance does not suffer greatly for the variance kernel. The other kernels do not perform well on this dataset.

The 1000 feature subset selected by the variance kernel outperformed the full dataset at low numbers of features; the performance achieved below 32 features is greater than the performance at the same operating point obtained with the full dataset. Given this performance, a satisfactory operating point at 16 features or even 8 features per class may be chosen, resulting in a very sparse predictor.

In summary, these results show that unsupervised pre-filtering does not degrade the classification performance and can improve the performance at few features. The RBF and variance kernels perform very well across both two-class datasets, with the other kernels not performing as consistently. On the multiclass



(a) Aggressive reduction to 50 features (b) Reduction to 500 features

Figure 5.2: Colon cancer dataset with centroid classifier and feature selector. ϵ-0 bootstrap with 200 repetitions. Error bars show 95% confidence interval. The experiment is identical to Figure 5.1, except with a different dataset.

CUP dataset, the variance kernel is the only kernel that performed well. The aggressive feature reduction – down to 50 features for the two-class datasets and 100 features for the CUP dataset – showed surprisingly good performance, suggesting that the full datasets contains significant redundancy and can be highly compressed without significant loss of performance.

van 't Veer in detail

To gain a better understanding of the relation between features selected by UBHSIC, the feature subsets obtained on the van 't Veer data were visualised. Subfigure 5.4a shows the full unfiltered dataset projected down onto the first two principal components with each sample represented by a number. It is clear from the visualisation that sample 10 is an outlier, sitting far away from the other samples. Excluding this sample and reprojecting the data obtains the embedding shown in Subfigure 5.4b. Here, one can observe that the samples roughly form two groups separated mostly by the first principal component (x-axis).

Subfigure 5.5a displays a biplot (Gabriel, 1971) of the dataset filtered down to 100 features using the linear kernel and UBHSIC. In the figure, samples are shown as black points and features as red vectors. It is clear here that the two-group structure observable on the original projection (Subfigure 5.4b) is maintained. Furthermore, the selected features are strongly positioned along the first principal

component. This is not unexpected as the linear kernel results in features being selected independently, and hence selects highly correlated feature sets. Indeed, a selection of 100 features most correlated with the first principal component yields a subset of features with 77 features in common with the subset selected by the linear kernel and UBHSIC.

The biplot produced using the RBF kernel (Figure 5.6) resembles the linear kernel results in that the two-group structure is preserved with many features selected along the first principal component. However, in comparison the features are more spread out in two fan-like structures, each spanning one of the groups well, whereas the "fans" formed by the linear kernel are not as spread out and well aligned with the groups.

Running the same analysis using the polynomial filter of degree 2 yields the results shown in Figure 5.6. Interestingly, the selected feature subset appears to have generated an outlier that is clearly visible in Subfigure 5.6a; removing this outlier produces a vastly different projection as shown in Subfigure 5.6b. In this figure, the feature vectors have a "radial" pattern, indicating the features did not have as high a cross-correlation as the previous kernels.

Finally, the variance kernel is shown in Figure 5.7. Unlike the polynomial kernel, the variance kernel did not produce any new outliers and resulted in a much more "radial" pattern than the polynomial filter. This indicates that the selected features were highly decorrelated as postulated previously.

These results indicate the linear and RBF kernels produce subsets containing cross-correlations; the linear kernel is especially highly cross-correlated and aligned with the first principal component while the RBF kernel spans the samples well and is less cross-correlated. The polynomial kernel and variance kernels result in much more decorrelated results, with the variance kernel producing highly decorrelated selections. Given the classification performance observed on the van 't Veer datasets, the RBF and variance kernels are both good choices and can be selected depending on one's preference for decorrelation.

5.3.2 Plant Genomics

A sugarcane dataset produced using DArT technology was evaluated using UBHSIC. This dataset consists of 55296 features and 80 plants. Each feature is measured on a continuous scale and may contain missing values. Each feature was scaled to have a mean of 0 and unit variance, with missing values subsequently set to 0. Features with no variance were removed before further analysis. The

aim is to reduce the number of features to 6972 so the samples can be rearrayed using a single microarray plate. Although the target number of features was 6972, the selection of a very small set (100, as in the van 't Veer data) was examined to simplify visualisation; with 6972 features the location of the points in the biplots will be obscured by the feature vector projections. Finally, note that there is a preference for decorrelation, as there are expected repeats among the initial 55296 features.

The same analysis as on the van 't Veer dataset was performed, starting with a projection of the full dataset onto the first 2 principal components (Figure 5.8). Unlike the van 't Veer dataset, no obvious initial outliers are present.

The visualisation of 100 features selected by UBHSIC using the linear kernel is shown in Figure 5.9. From Subfigure 5.9a it is clear the linear kernel has resulted in two outliers, which after removal produce the biplot shown in Subfigure 5.9b. As on the previous datasets, the linear kernel has produced selections aligned well with the first principal component. Figure 5.10 shows the results from filtering using the RBF kernel, and like the previous datasets the selected features are polarised along the first principal component, though less so than for the linear kernel; some of the selected features are aligned better with the 2^{nd} principal component. Overall, the RBF kernel appears to span the samples better than the linear kernel.

Figure 5.11 shows the biplot resulting from filtering using a polynomial kernel of degree 2. Again outliers are generated, but once removed the resulting biplot shows the radial pattern rather than the fan pattern indicating the features selected are decorrelated to some extent.

Finally, Figure 5.12 shows the biplot obtained after filtering using the variance kernel. Again, one can observe the radial pattern (more so than the polynomial kernel) indicating highly decorrelated selections. These results concur with the results of the previous section, and demonstrate that the variance kernel can produce decorrelated selections, which is desired in this context.

Two heatmaps were generated and compared from the reduced datasets obtained using the RBF and variance kernels. For each kernel, a dataset of 7000 features was selected and heatmaps generated using the standard Euclidean distance measure and agglomerative hierarchical clustering (Hastie et al., 2001). Figure 5.13 and Figure 5.14 show the heatmaps after filtering using the variance and RBF kernels. The decorrelated selections observed in the biplot for the variance kernel can be seen reflected as less visible structure in the heatmap when

compared to the RBF filtered results.

5.3.3 Quantum vs Simulated Annealing

The proposed quantum annealing optimisation algorithm was compared against the more widely known and used simulated annealing algorithm (Belisle, 1992) on the van 't Veer and sugarcane datasets. The variance kernel was used on both datasets, and each calculation of the energy function was logged. Both algorithms were run using the same annealing schedules, and the quantum annealing algorithm was constrained to between 4 and 10 particles (inclusive).

Figure 5.15 shows the results for both datasets. In both cases, the quantum annealing algorithm converged to a lower energy state than the simulated annealing algorithm. Increasing the maximum number of particles for the quantum annealing algorithm to 200 does not significantly change the results; quantum annealing still converges to a lower energy state than simulated annealing in a shorter time frame.

5.4 Conclusions

A method for unsupervised feature selection, UBHSIC, based on the HSIC was presented and evaluated on several bioinformatics datasets from cancer genomics and plant genomics. The results are very promising; on the cancer genomics datasets the classification performance was increased after filtering the initial dataset down to a few features. On the sugarcane dataset, a highly decorrelated subset was obtained as desired for rearraying.

The flexibility of this method provided by the kernels is very attractive. It allows tailoring towards a given problem, and many existing kernels may be used. Furthermore, as UBHSIC scales with $O(n^3)$ where n is the number of samples, it is suitable for the high-dimensional low-sample datasets frequently encountered in bioinformatics. In all datasets tested, the proposed variance kernel performed well. Thus, it is a good baseline choice in cases where the choice of kernel is unclear.

Algorithm 5.1 UBHSIC procedure, part 1 (initialisation)

procedure UBHSIC($\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m, k\colon \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}, n_{\text{feats}}, n_{\text{iter}}$)
  function HSIC($\mathbf{S}$) ▷ The objective function
    $\theta \leftarrow \{i|S_i = 1\}$
    $K'_{ij} \leftarrow k((x_{if})_{f\in\theta}, (x_{jf})_{f\in\theta})$ ▷ The reduced dataset kernel
    return $-\operatorname{tr}(K'HKH)$
  end function

  function DIFFUSE($\mathbf{S}, \gamma$) ▷ Diffuses a particle
    $i \leftarrow \text{sample}(\{i|S_i = 1\})$ ▷ Randomly select a positive index
    $j \leftarrow \text{sample}(\{j|S_j = -1\})$ ▷ Randomly select a negative index
  Diffuse particle:
    $\mathbf{S}' \leftarrow \mathbf{S}$
    $S'_i \leftarrow -1$
    $S'_j \leftarrow 1$
    if HSIC($\mathbf{S}'$) < HSIC($\mathbf{S}$) **then** ▷ Accept if lower energy is achieved
      return $\mathbf{S}'$
    else if RAND < $\exp(-(\text{HSIC}(\mathbf{S}') - \text{HSIC}(\mathbf{S}))/\gamma)$ **then** ▷ accept with probability determined by the Boltzmann distribution and the current temperature γ
      return $\mathbf{S}'$
    end if
    return $\mathbf{S}$ ▷ Reject move
  end function

  Calculate constants:
  $H \leftarrow Id - \frac{1}{n}\mathbf{1}\mathbf{1}^*$
  $K_{ij} \leftarrow k(\mathbf{x}_i, \mathbf{x}_j)$

  Calculate initial particle position:
  $\mathbf{o} \leftarrow \text{ORDER}(X^*HKHX)$ ▷ order indices in decreasing order
  $\mathbf{S} \leftarrow -1$
  for $i \in \{1, \ldots, n_{\text{feats}}\}$ **do**
    $S_{o_i} \leftarrow 1$
  end for
  $pool \leftarrow \{\mathbf{S}\}$

Continued in Algorithm 5.2

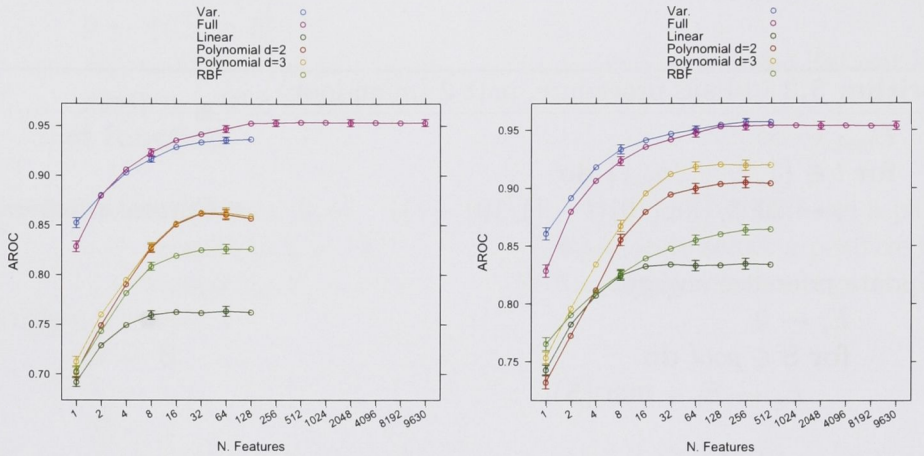
Algorithm 5.2 UBHSIC procedure, part 2 (main loop)

$E_{\text{best}} \leftarrow \infty$
for $t \in \{1, 2, \ldots, n_{\text{iter}}\}$ **do**
  $\gamma \leftarrow 2000/(\log(10\lfloor(t-1)/10\rfloor + e))$ ▷ Current temperature

Update reference energy:
  $E_r \leftarrow 0$
  for $\mathbf{S} \in pool$ **do**
    $E_r \leftarrow E_r + \text{HSIC}(\mathbf{S})$

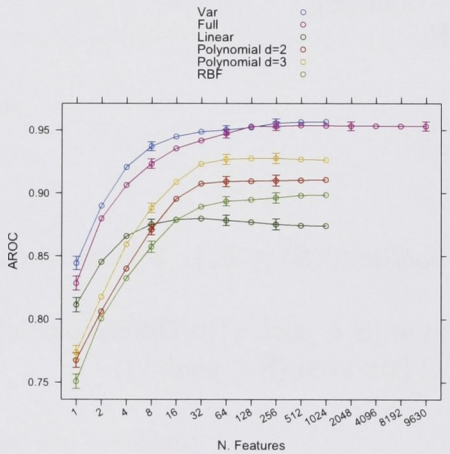
Track best solution
    if $\text{HSIC}(\mathbf{S}) < E_{\text{best}}$ **then**
      $E_{\text{best}} \leftarrow \text{HSIC}(\mathbf{S})$
      $\hat{\mathbf{S}} \leftarrow \mathbf{S}$
    end if
  end for
  $E_r \leftarrow E_r/|pool|$

Update pool:
  $pool \leftarrow \{\mathbf{S} \in pool | \text{HSIC}(\mathbf{S}) \leq E_r\}$ ▷ Discard particles above reference energy
  $pool' \leftarrow \{\text{DIFFUSE}(\mathbf{S} \in pool, \gamma)\}$ ▷ Diffuse particles creating a new pool
  $pool' \leftarrow pool' \cup \{\text{DIFFUSE}(\mathbf{S} \in pool, \gamma)\}$ ▷ Duplicate particles
  $pool \leftarrow pool'$
end for
return $\{i | \hat{S}_i = 1\}$ ▷ Return best feature set found
end procedure



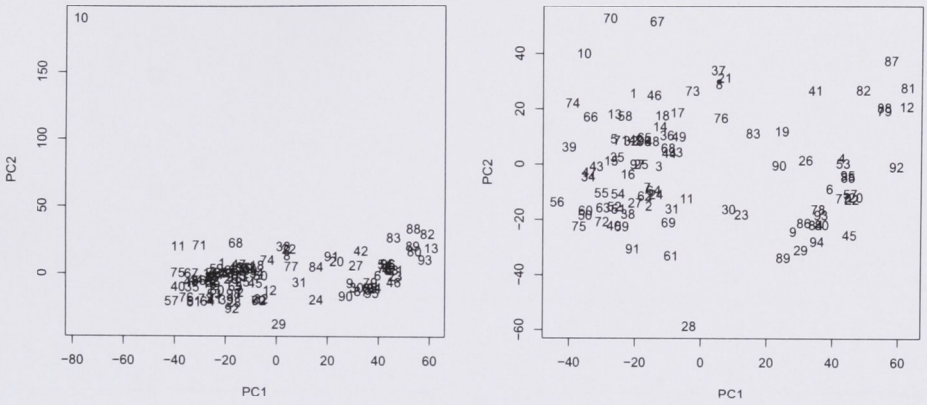
(a) Reduction to 100 features

(b) Reduction to 500 features



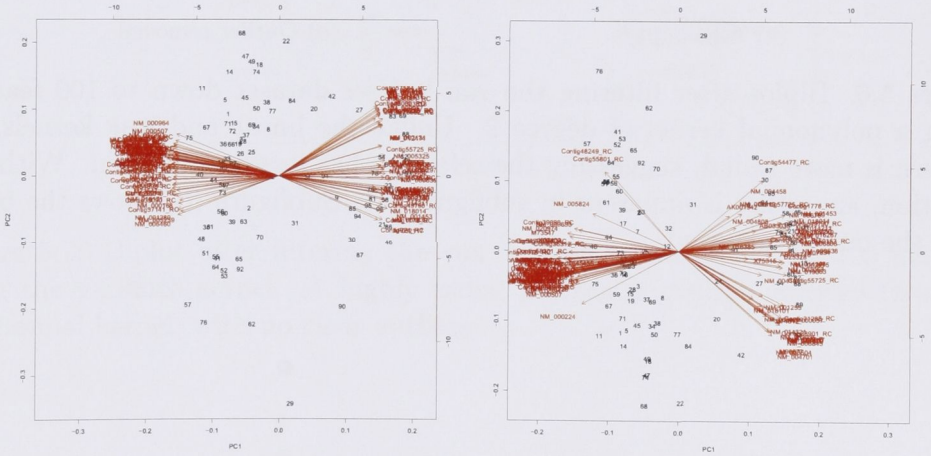
(c) Reduction to 1000 features

Figure 5.3: CUP cancer dataset with centroid classifier and feature selector. ϵ-0 bootstrap with 200 repetitions. Error bars show 95% confidence interval. Number of features shown is per class, not overall. Experiment details are as in Figure 5.1.



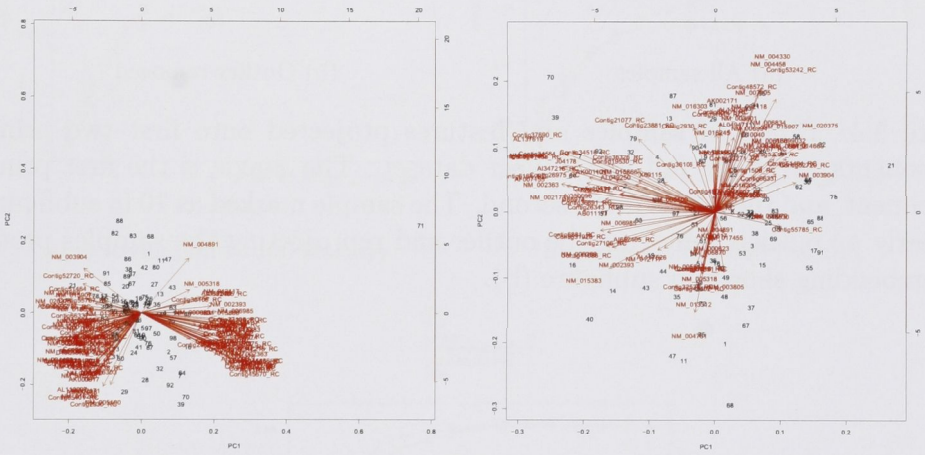
(a) All samples (b) Outlier removed

Figure 5.4: Biplot of samples and features projected onto first two principal components using the full van 't Veer dataset. The x-axis is the first principal component, and the y-axis is the second. The sample marked as 10 in subfigure (a) is clearly an outlier; removing the outlier and reprojecting the samples produces the embedding shown in subfigure (b).



(a) Linear kernel (b) RBF kernel

Figure 5.5: Biplot after filtering the van 't Veer dataset down to 100 features using the linear and RBF kernels. Both kernels produce selections polarised along the first principal component, though the RBF kernel selections span the samples better than the linear kernel selections.



(a) All samples (b) Outlier removed

Figure 5.6: Biplot after filtering the van 't Veer dataset down to 100 features using a polynomial kernel of degree 2. Unlike the linear and RBF kernels, the pattern is more radial, suggesting the selection has less coregulation. With this selection, an outlier is apparent in subfigure (a). Subfigure (b) shows the biplot with the outlier removed.

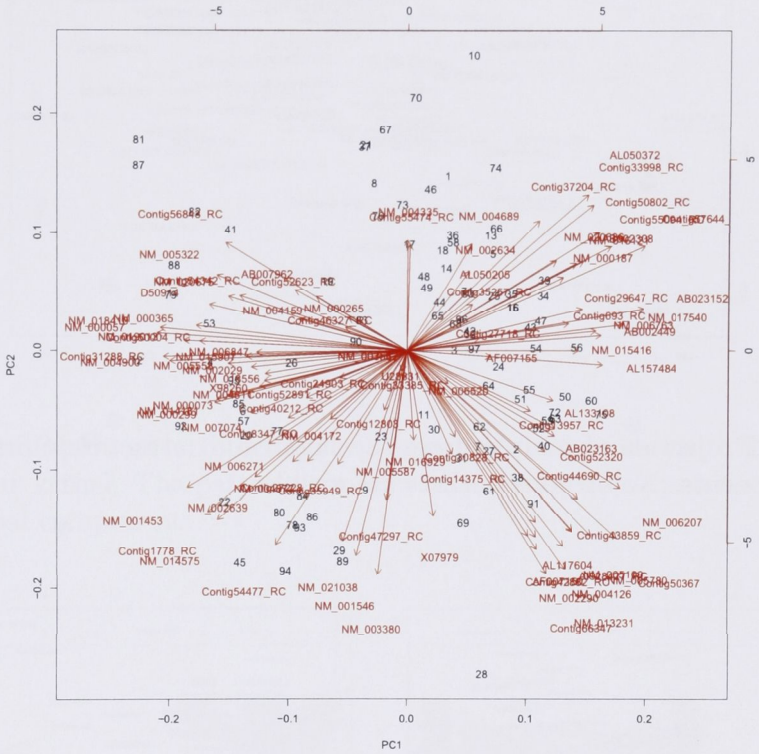


Figure 5.7: Biplot after filtering the van 't Veer dataset down to 100 features using the variance kernel. A highly radial pattern is visible, more-so than the polynomial kernel, with no clear outliers.

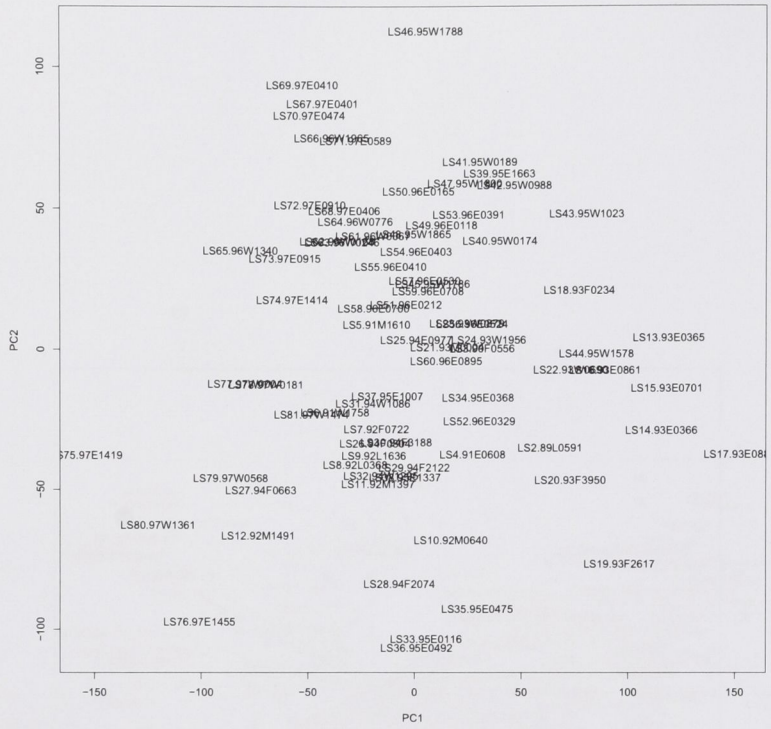
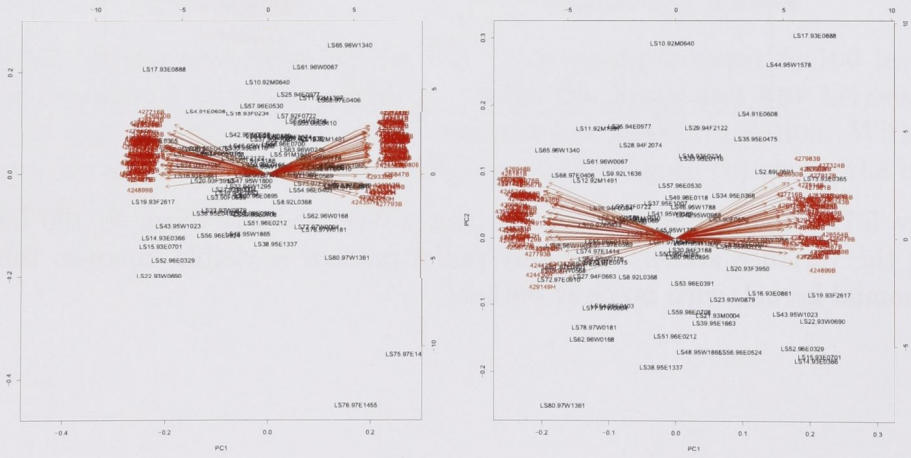


Figure 5.8: Projection of samples from the sugarcane dataset onto first two principal components. No obvious outliers are visible.



(a) All samples (b) Outliers removed

Figure 5.9: Biplot after filtering the sugarcane dataset down to 100 features using a linear kernel. Similarly to the van 't Veer dataset, the selections are polarised along the first principal component. Two outliers are visible in subfigure (a); subfigure(b) shows the recalculated biplot with the outliers removed.

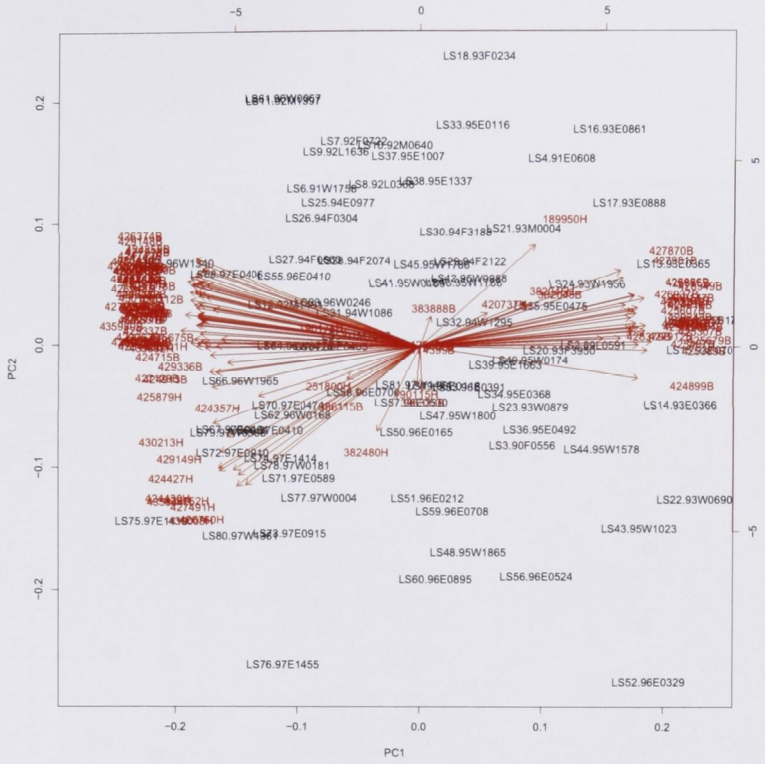
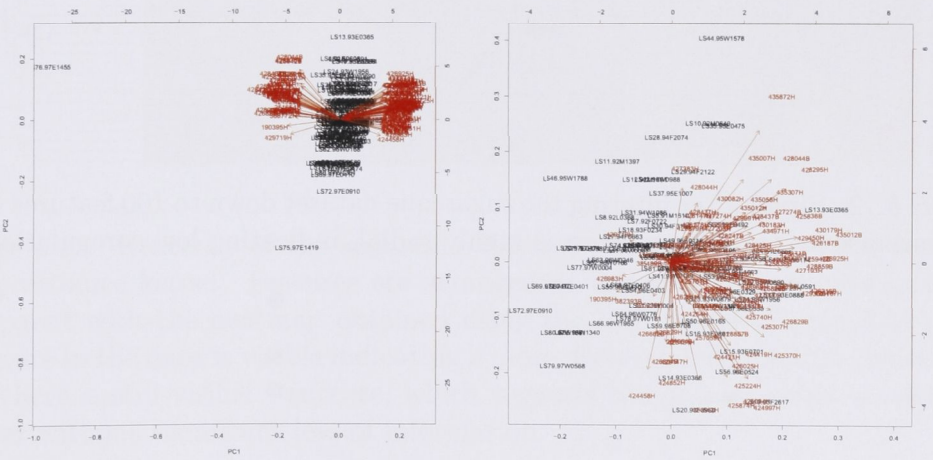


Figure 5.10: Biplot after filtering the sugarcane dataset down to 100 features using an RBF kernel. The pattern is more radial, though still polarised along the first principal component.



(a) All samples (b) Outliers removed

Figure 5.11: Biplot after filtering the sugarcane dataset down to 100 features using a polynomial kernel of degree 2. Two outliers are clearly visible in subfigure (a), and a radial pattern is visible after outlier removal in subfigure (b).

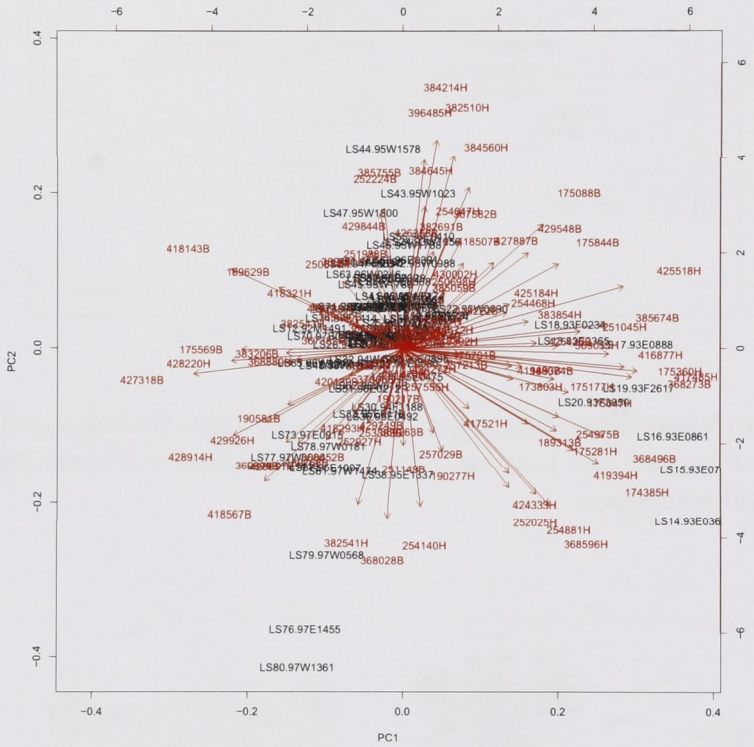


Figure 5.12: Biplot after filtering the sugarcane dataset down to 100 features using the variance kernel. The pattern is highly radial, indicating low coregulation.

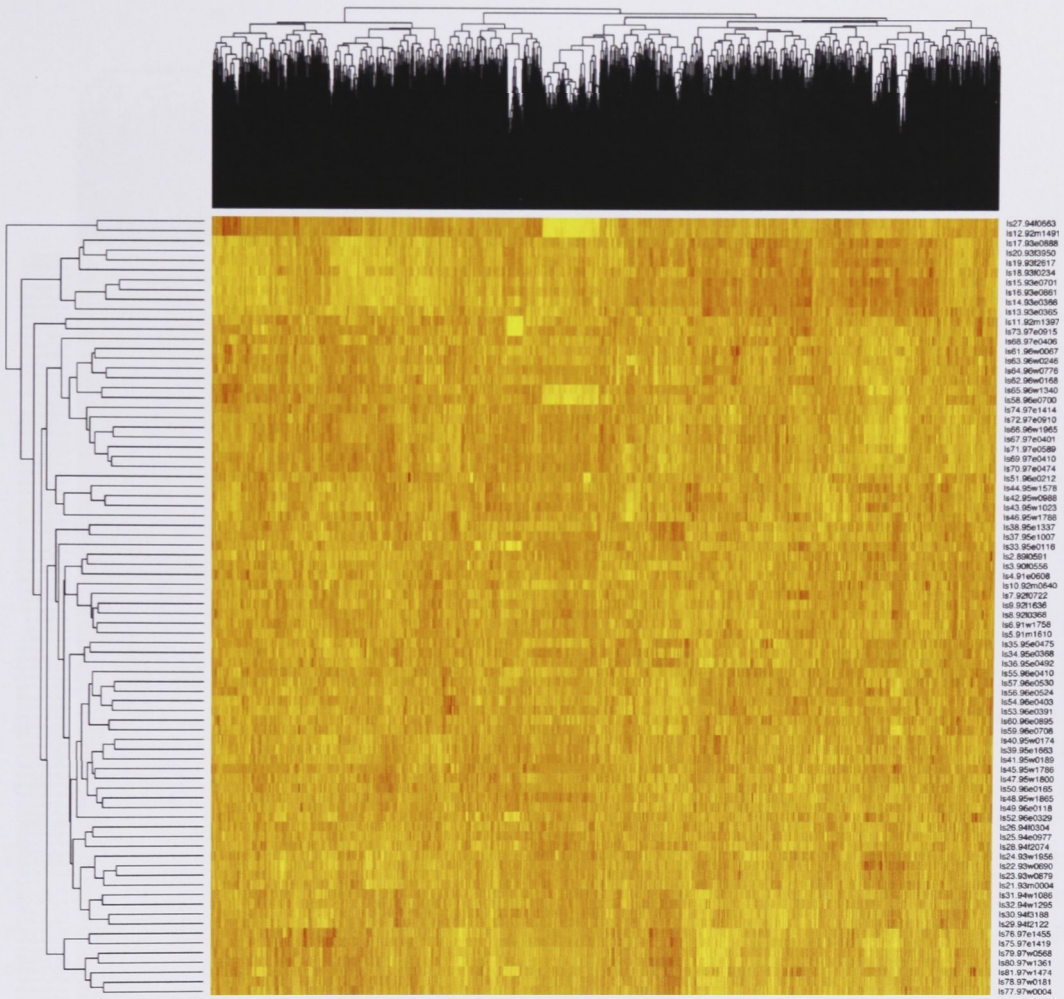


Figure 5.13: Heatmap of the sugarcane dataset filtered down to 7000 features using variance kernel. Features are arranged as columns and samples as rows. Compared with the heatmap obtained using the RBF kernel (Figure 5.14), less structure in the data is visible indicating a more decorrelated selection. Nevertheless, there is still visible structure, which suggests the dataset could be reduced further without significant loss of information.

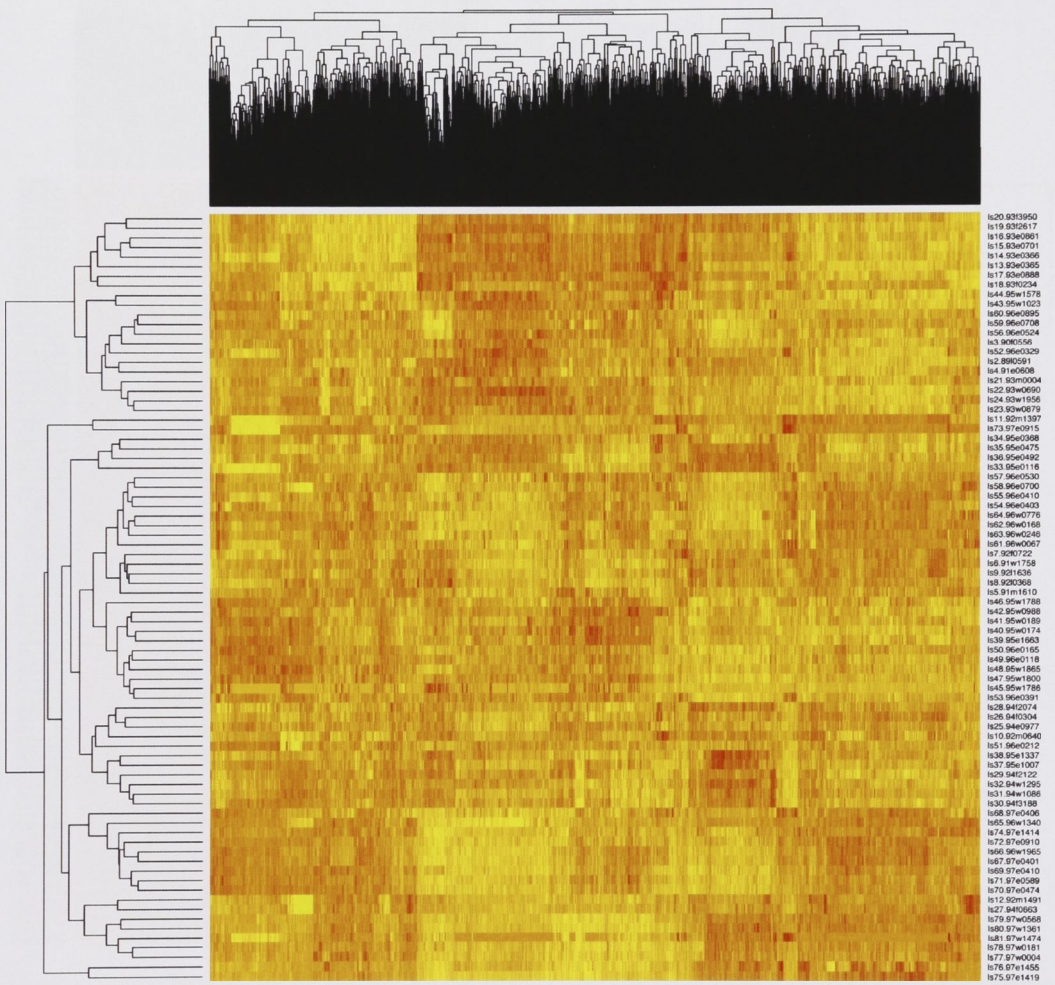
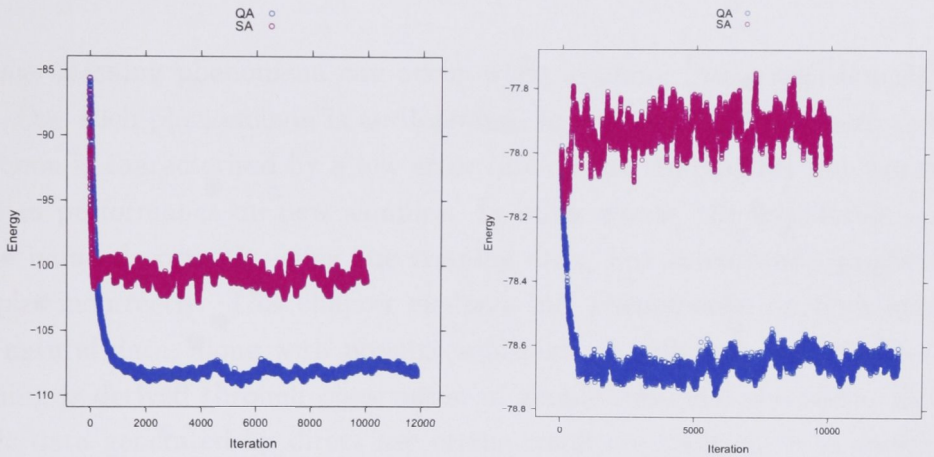


Figure 5.14: Heatmap of the sugarcane dataset filtered down to 7000 features using the RBF kernel. Features are arranged as columns and samples as rows. Compared to the heatmap obtained using the variance kernel (Figure 5.13), significantly more structure is visible indicating higher covariance.



(a) van 't Veer (b) Sugarcane

Figure 5.15: Quantum annealing compared against simulated annealing on the van 't Veer and sugarcane datasets using the variance kernel. The energy after every call to the objective function is plotted with the quantum annealing calls shown in blue and the simulated annealing calls shown in purple. Quantum annealing reaches a lower energy state than simulated annealing on both datasets.

Chapter 6

Antilearning

Strange learning phenomena can occur when learning from small-sample datasets. One such phenomenon is antilearning. In two-class classification, this phenomenon is characterised by a low error rate on the training set but worse than random performance on new samples. In other words, the hypothesis appears to be induced correctly using the training data, but *consistently* predicts new samples incorrectly. This chapter explores this phenomenon on both synthetic and natural data, along with possible solutions. A sufficient condition for antilearning is derived through observation of simple two-class classifiers, and synthetic data generated by direct use of the condition created. It is shown that feature selection does not extract learnable behaviour from the data. A method of detecting the data's mode (antilearnable/learnable) and reversing the classification rule if necessary, based on a sufficient condition for antilearning, is presented and evaluated on both synthetic and natural data.

6.1 Motivational Examples

Consider first the X-OR problem in two dimensions illustrated in Figure 6.1. Without loss of generality suppose the leftmost point is removed for testing and the remaining points are used for training. Using either the centroid or SVM classifiers with a linear kernel yields the decision hyperplane shown. This hyperplane perfectly classifies the training data, but incorrectly classifies the withheld point. In this case, the "fix" for antilearning is simple – a non-linear classifier can fit the data and learn normally, though not all non-linear classifiers can solve the

X-OR problem; a k-nearest neighbours (KNN) classifier[1] with the normal Euclidean metric will anti-learn.

A sufficient condition for antilearning can be derived by studying Rosenblatt's perceptron algorithm (see Algorithm 2.1). Given a training set $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^m \times \{\pm 1\}$ (which is assumed to be centred), there exists $\boldsymbol{\alpha} \in \mathbb{N}^n$ such that the Rosenblatt perceptron solution is given by $\boldsymbol{\beta} = \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i$ with the hypothesis $f\colon \mathbb{R}^m \to \mathbb{R};\ \mathbf{x} \mapsto \langle \mathbf{x}, \boldsymbol{\beta} \rangle$. Suppose $(\mathbf{x}, y) \notin \mathcal{X}$ is available for testing. For the sample to be misclassified,

$$y \left\langle \sum_{i=1}^n y_i \alpha_i \mathbf{x}_i, \mathbf{x} \right\rangle = y \sum_{i=1}^n y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle = \sum_{i=1}^n y y_i \alpha_i \langle \mathbf{x}', \mathbf{x} \rangle < 0 \tag{6.1}$$

and therefore $E_P[yy' \langle \mathbf{x}', \mathbf{x} \rangle] < 0$ is a sufficient condition for antilearning, where P is the underlying probability measure generating the data.

It follows that if $y = y'$, then for the condition to hold $\langle \mathbf{x}', \mathbf{x} \rangle < 0$, and similarly $\langle \mathbf{x}', \mathbf{x} \rangle > 0$ if $y \neq y'$. Intuitively this suggests that samples from opposite classes are *more similar* than samples of the same class. This condition clearly holds for the X-OR problem, so Rosenblatt's perceptron will antilearn on this example.

This sufficient condition immediately suggests a method for generating synthetic antilearning data by simply generating a cloud of random points and assigning labels to minimise the objective function

$$\sum_{(\mathbf{x},y),(\mathbf{x}',y') \in \mathcal{X}} yy' \langle \mathbf{x}', \mathbf{x} \rangle .$$

This method will be called the *direct sufficient condition* (DSC) method. Another method that yields antilearnable data is the *orthogonal frame projection* (OFP) method and was studied by Kowalczyk et al. (2007). Given a set of random points $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ such that $m > n$, the points are orthogonalised so $\langle \mathbf{x}_i, \mathbf{x}_j \rangle = 0\ \forall i, j$. A random hyperplane $\boldsymbol{\omega}$ of dimension $m - 1$ is then generated, and the samples projected onto the hyperplane, producing the set of points $\mathbf{x}_i' = \boldsymbol{\omega} \mathbf{x}_i$. The labels $y_i \in \{\pm 1\}$ are assigned depending on which side of the hyperplane the original points were. This produces an antilearnable dataset $\mathcal{X} = \{(\mathbf{x}_i', y_i)\}_{i=1}^n$.

[1] KNN classifiers predict unknown samples as the majority class of the k nearest neighbours, typically measured using the Euclidean distance.

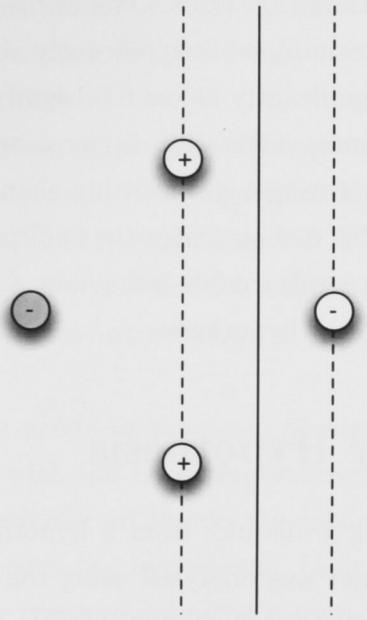


Figure 6.1: X-OR antilearning example. Grey point withheld as an independent test, resulting in decision hyperplane which perfectly antilearns the dataset.

6.2 Analysis of Synthetic Data

Several different classification and feature selection methods were evaluated on synthetic data to measure the extent of antilearning. Of the set of classifiers, the centroid (Chapter 4), perceptron (Algorithm 2.1), shrunken centroid (PAM, see Section 4.7.2), RFE-SVM (Section 2.5.2), and SVM classifiers (Proposition 2.24) have been introduced previously. Two more non-linear classifiers, random forests (Hastie et al., 2001) and k-nn (Hastie et al., 2001), were were also evaluated. The k-nn algorithm is a simple classifier, predicting new samples as the majority class of the k closest sample in the training data. Random forests are an extension of decision trees, whereby many decision trees are created by splitting each node on a selection of random features. A full description of random forests can be found elsewhere (Hastie et al., 2001). For the non-embedded methods (SVM, centroid, random forests, and k-nn), the CFS and t-test feature selection filters were used to evaluate the effect of feature reduction.

Figure 6.2 shows the results of an analysis using the various algorithms on a dataset generated by DSC. The dataset contained 100 samples, 57 in the negative class and 43 in the positive class, each consisting of 1000 dimensions. The ϵ-0 bootstrap and AROC metric (Definition 2.9) with 25 bootstrap repetitions was

used to estimate the generalisation error. The antilearning effect is quite clear; the AROC achieved on the training set is significantly above 0.5, while the AROC on the withheld samples is significantly below 0.5, signifying the hypothesis is consistently incorrect. The linear separating hyperplane methods (SVM, centroid, Perceptron) demonstrated stronger antilearning than the non-linear techniques (random forests, KNN). This demonstrates the sufficient condition for antilearning (Equation 6.1) causes antilearnable behaviour for a large class of learning machines, including non-linear hypothesis.

6.3 Non-Linear Hypothesis

To verify that antilearning is distinct from a hypothesis class with insufficient power, the synthetic dataset was analysed using the centroid classifier and the Gaussian RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma||\mathbf{x} - \mathbf{x}'||_2^2)$ for various settings of the hyperparameter σ. Figure 6.3 shows the AROC results of an ϵ-0 bootstrap with 25 repetitions experiment on the same synthetic dataset as previously. Here it is clear that though learning behaviour can be extracted for extremely small values of σ, the levels are very low and close to random guessing. In contrast, the maximum level of antilearning achieved ($\approx$ 10% AROC) was far in excess of the maximum level of learning. Consequently, the conclusion is antilearning for the synthetic dataset is not induced by an insufficient hypothesis class, but is rather inherent in the structure of the data.

6.4 Reversible Learners

Consider the task of learning from antilearnable datasets. Given that antilearning is characterised by consistent misclassification of new samples, reversing the hypothesis should produce good predictions for new samples. The question, however, is can the decision to reverse a classifier be made in a principled rather than ad-hoc way.

The first reversible algorithm was originally presented by Kowalczyk (2007a) and is relies on simple "brute force" detection of antilearnable structures. In essence, a resampling method is used on the training data to estimate the AROC for new samples and the classifier reversed if the estimated AROC is below 0.5. Algorithm 1 shows the *leave-one-out reverser* (LOO-Rev) algorithm implementing this method using LOO estimation on the training data. This algorithm is

particularly attractive when using the centroid algorithm as the LOO-AROC estimate can be calculated cheaply (see Section 4.5). An explicit description of this algorithm is given in Algorithm 6.1.

Another method is to directly use the sufficient condition for antilearning to determine if reversing is required. This algorithm is called the *kernel reverser* (k-Rev) and is explicitly outlined in Algorithm 6.2. This method is clearly faster than the LOO-Rev method, however it has the restriction that D in Algorithm 6.2 must be evaluated before any feature selection as feature selection can break the antilearning structure.

Another restriction one needs to be aware of when using these reversers is that they cannot be used with the bootstrap estimator as the high number of replicated samples in the training set breaks the antilearning geometry and thus the algorithms will not reverse the hypothesis. Consequentially, the replicated bootstrap samples must be removed from the training set before inducing a hypothesis. Note that this is still different from a repeated holdout as the size of the training set varies stochastically.

Figure 6.4 shows the results of an ϵ-0 bootstrap, with no replicates in the training set, for the centroid classifier using both reverser methods on the same synthetic data as previously. Again, 25 bootstrap repetitions were used and the AROC metric evaluated. The k-Rev reverser (Algorithm 6.2) works very well, producing an exact inverse curve to the centroid method. The LOO-rev (Algorithm 6.1) also resulted in a curve in the learning region, however the variance was increased and the performance achieved was lower than the k-Rev curve. It is not surprising that the k-Rev method performs so well as the dataset was generated to satisfy the antilearning condition. Likewise, the LOO-rev is expected to perform worse as the LOO estimate of AROC is noisy and inaccurate. However, both methods resulted in a hypothesis clearly positioned in the learning region ($> .5$ AROC).

6.5 Natural Antilearning

Antilearning also arises in natural data; one such dataset is the *adenocarcinoma* dataset (Greenawalt et al., 2007). This dataset is part of an oesophageal cancer patient study of 46 cancer patients; 25 were adenocarcinoma patients and 21 were squamous cell carcinoma (SCC). Each patient was measured for 9857 gene expression levels using microarrays. The goal of the study was to produce a

Algorithm 6.1 LOO-Rev

Let:

1. $\theta\colon 2^{\mathbb{X}\times\mathbb{Y}} \to \mathbb{R}^{\mathbb{X}}$ be a hypothesis inducer;
2. $\mathcal{X} := \{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{X} \times \mathbb{Y}$ be a training dataset;
3. $$\text{AROC}\colon 2^{\mathbb{Y}\times\mathbb{R}} \to [0,1]$$
$$(\{(y_i, y_i')\}_{i=1}^n) \mapsto P(y_i' > y_j' | y_i > y_j) + \frac{1}{2} P(y_i' = y_j' | y_i > y_j)$$
be a function measuring the AROC given a set of labels and a set of continuous predictions.

1: $f' \leftarrow \theta(\mathcal{X})$
2: **for** $i = 1, \ldots, n$ **do**
3: $\quad f_i \leftarrow \theta(\mathcal{X} \setminus (\mathbf{x}_i, y_i))$
4: **end for**
5: $D \leftarrow \text{AROC}(\{(y_i, f_i(\mathbf{x}_i)\}_{i=1}^n)$
6: **return** $f = \text{sign}(D - 0.5) f'$

Algorithm 6.2 k-Rev

Let the assumptions of Algorithm 6.1 hold, with the additional constraint that the data is centred.

1: $D \leftarrow \sum_{(\mathbf{x},y)\in\mathcal{X}} \sum_{(\mathbf{x}',y')\in\mathcal{X}\setminus\{(\mathbf{x},y)\}} yy' \langle \mathbf{x}, \mathbf{x}' \rangle$
2: **return** $f = \text{sign}(D) f'$

2-class classifier for response to chemotherapy capable of separating patients into "good" and "poor" responders. Interestingly, the classifiers induced on the SCC subgroup learned normally, but the classifiers induced on the adenocarcinoma patients antilearned. This difference has been previously published (Greenawalt et al., 2007) and here the focus will be on learning from the antilearable adenocarcinoma subset.

Figure 6.5 shows the results of the ordinary centroid and the centroid with the reverse algorithms applied to it. Like the simulated data, the reversers were able to detect the antilearning structure and correctly reverse the classifier. Like the synthetic data, again one can observe the k-Rev algorithm outperforming the LOO-Rev algorithm.

As a positive control, the same experiment was conducted on the colon cancer and van 't Veer datasets studied in the previous chapter. As this is not an

antilearning dataset, all three curves should sit within the learning region (> 0.5) if the reverse correctly detects the learnable structure. If the reverser algorithms incorrectly determine the mode of the data, the reverser curves will sit in the antilearning region (< 0.5). Figure 6.6 shows the results of the experiment. On the colon cancer dataset, every plot is identical and above level of random guessing, signifying that both the k-Rev and LOO-Rev algorithms were able to detect the learning structure easily. The van 't Veer dataset is not quite as clean, with the LOO-Rev not producing an identical curve (but producing a curve with no significant difference). This is likely due to the van 't Veer dataset having an overall weaker signal.

A natural question that arises is how significant are the antilearning results given the extremely small sample size (25 samples). When dealing with small samples, apparent antilearning can be seen that is caused not by any inherent structure of the data, but by the resampling into training and test sets for estimation of generalisation error (Parker et al., 2007). This bias is caused by the negative correlation between the class balance in the training and test set. As an example, consider the perfectly balanced case where there are equal quantities of positive and negative samples in the dataset. The removal of one sample, say a positive sample, from this set for testing skews the class distribution in the training set towards the negative class, and the class distribution in the test set towards the positive class. A majority voter will then appear to exhibit antilearning behaviour as it will consistently predict any withheld test sample incorrectly.

Of course, this bias varies depending on the resampling strategy and the sensitivity of the classifier to class proportions. The example given above was for leave-one-out (LOO) cross-validation[2], which is one of the worst performers in terms of stratification bias (Parker et al., 2007). It should be noted that a stratified bootstrap[3] does not suffer from this particular stratification bias as the class proportions remain constant through every iteration. Finally, regardless of the choice of classifier or resampling technique, a simple label permutation test will determine if the results are significant, and will incorporate issues arising from this particular bias.

To this end, a label permutation test was conducted with 1000 permutations.

[2]The LOO estimator is particularly popular within the bioinformatics community as the common thought is that it is a more accurate estimator for small samples. However, LOO must be judiciously applied to avoid problems arising from this pessimistic bias.

[3]Training sets are sampled with replacement *per class* to preserve the class proportions across each iteration

Figure 6.7 shows the results of the permutation tests for the centroid and the k-Rev centroid. For the centroid classifier, the results lie outside the $p = 0.05$ significance threshold which suggests the signal is genuine and did not arise by random chance. Note that this distribution is clearly a skew-right distribution – the threshold for significance at the $p = 0.05$ level on the learning side is higher than on the antilearning side. This result is not too surprising and suggests antilearning datasets do not arise as easily as learning datasets, i.e., the particular geometry required to produce antilearning does not arise as frequently. The k-Rev permutation results are equally significant, with the unpermuted result well placed in the learning region with $p < 0.05$. Interestingly, the null distribution for the k-Rev algorithm has reduced the skewness easily observable in the null distribution for the centroid classifier. This would suggest that many random labellings have an antilearnable structure during training but not during testing, thus causing the k-Rev algorithm to flip the classifier into the antilearning region.

6.6 Conclusions

Antilearning is a real phenomenon that arises in natural bioinformatics datasets and can be synthesised easily by many different models. It was shown that the problem does not arise from an insufficient hypothesis class (i.e., non-linear predictors do not increase performance), and also cannot be improved by increasing regularisation as antilearning was present with the centroid classifier (see Chapter 4). Consistent detection and hence correction of antilearning is possible as demonstrated by the experiments with the LOO-Rev and k-Rev algorithms. The level of significance achieved both by the ordinary centroid operating in antilearning mode and the centroid corrected by the k-Rev algorithm operating in learning mode on the natural dataset was $p < 0.05$, suggesting the adenocarcinoma dataset possesses a real and antilearnable signal.

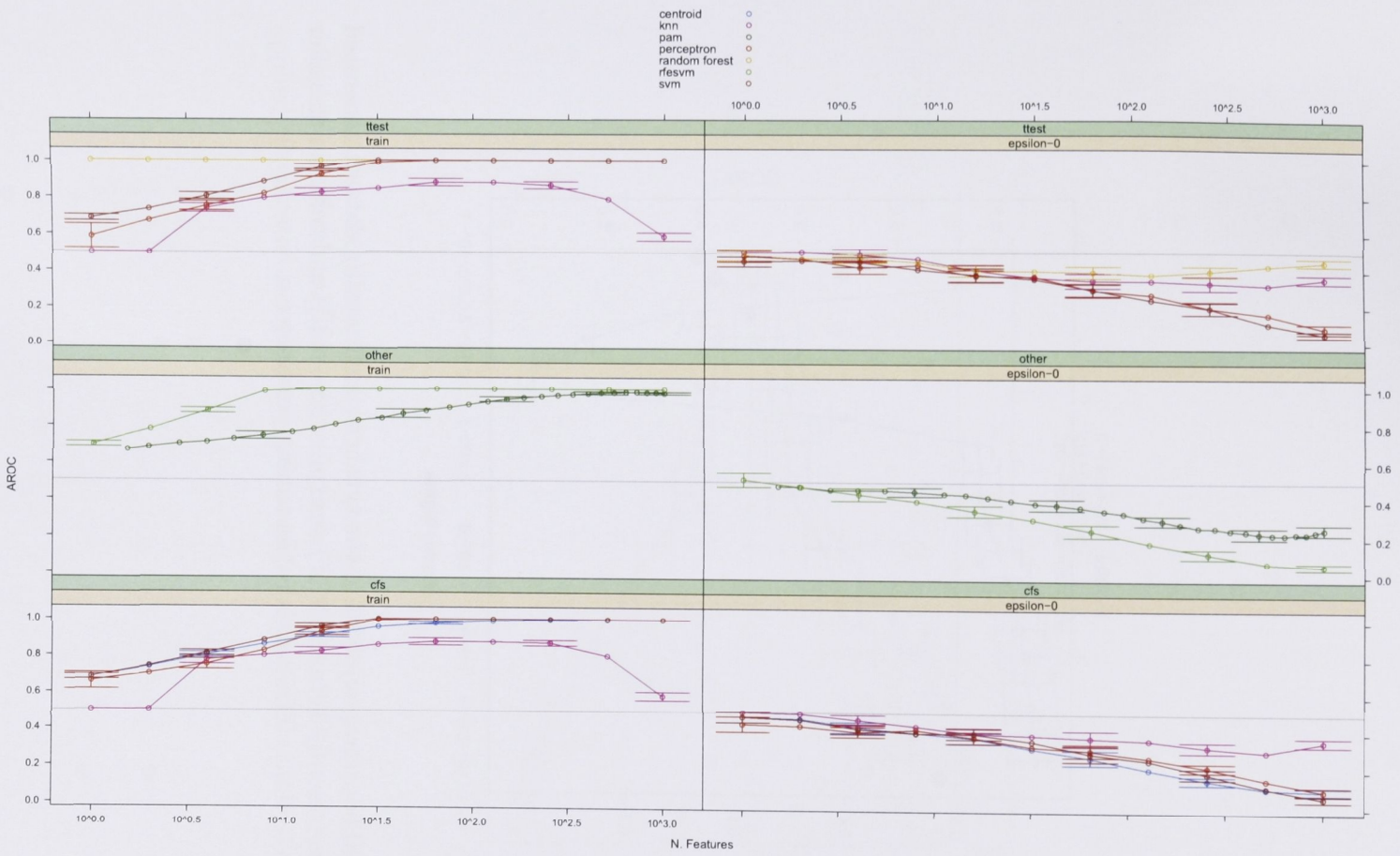


Figure 6.2: ϵ-0 bootstrap analysis of the synthetic antilearning dataset generated with DSC using the AROC metric and various combinations of feature selection methods and classification algorithms. 25 bootstraps were used. Bars indicate the 95% confidence interval for the mean. Antilearnable behaviour (high training accuracy, below random performance on the independent test set) is visible for every combination.

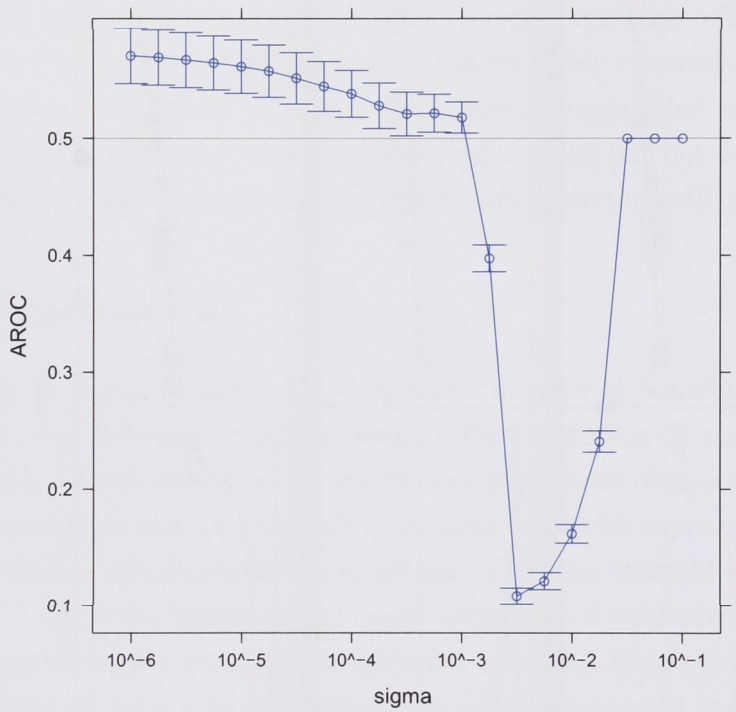


Figure 6.3: ϵ-0 bootstrap analysis of the synthetic antilearning dataset generated with DSC using the RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-\sigma||\mathbf{x}-\mathbf{x}'||_2^2)$ and centroid classifier (no feature selection). The x-axis is the tuning parameter σ.

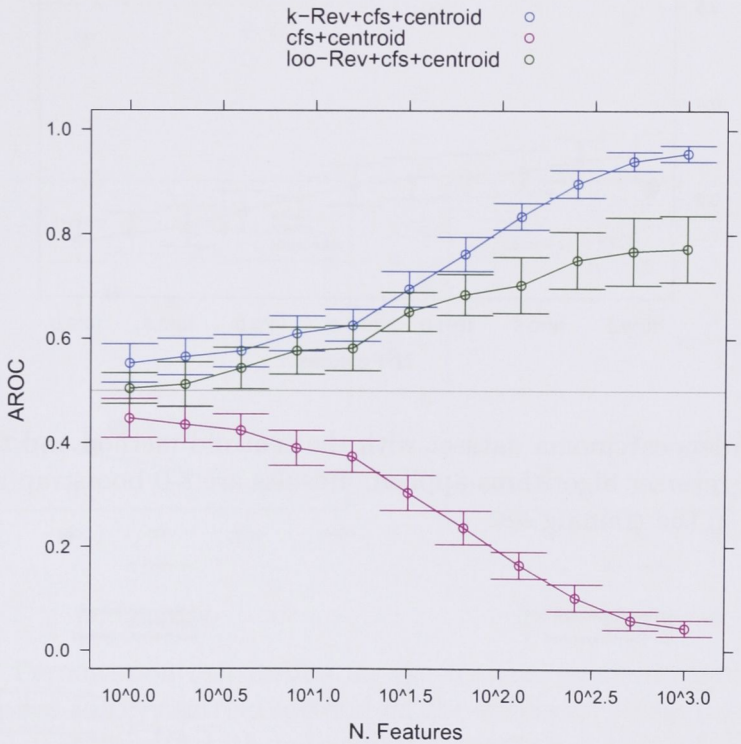


Figure 6.4: ϵ-0 bootstrap analysis of the synthetic dataset generated with DSC using the centroid method and the centroid method with reverser algorithms applied. Replicated samples were removed from training set.

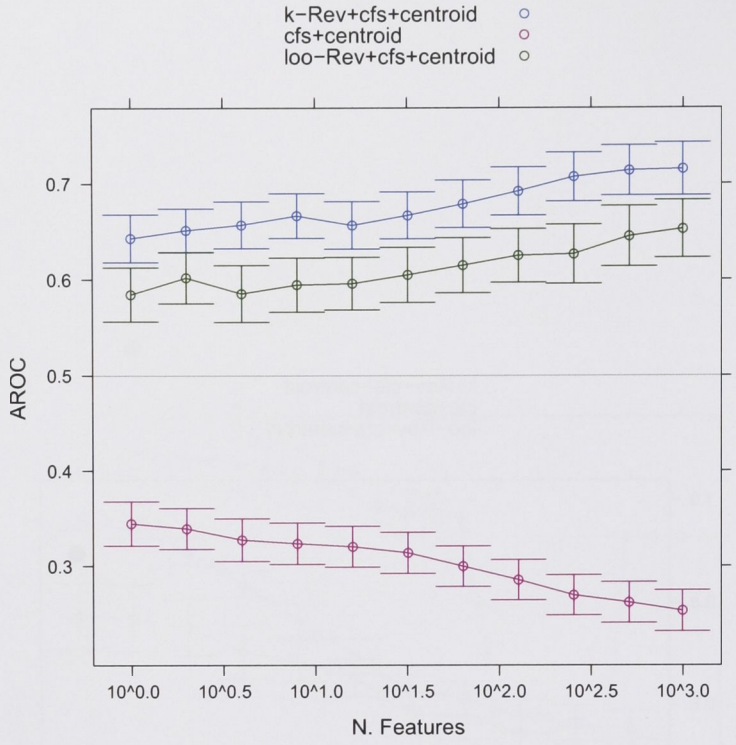
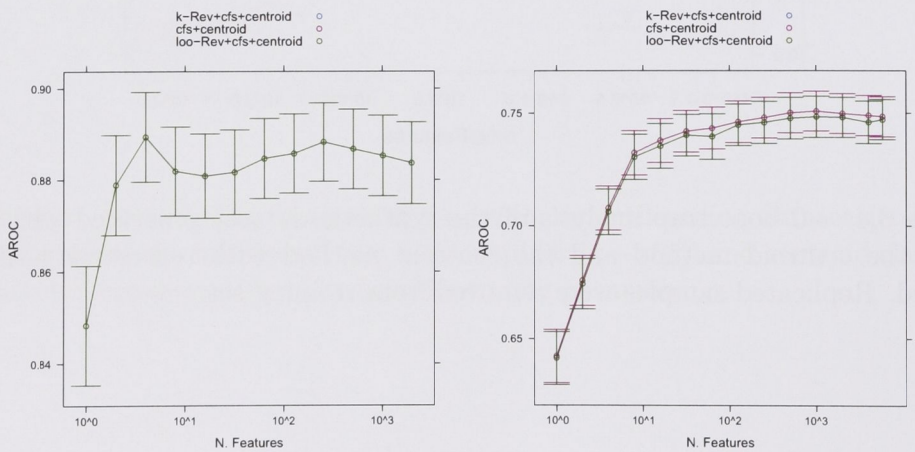
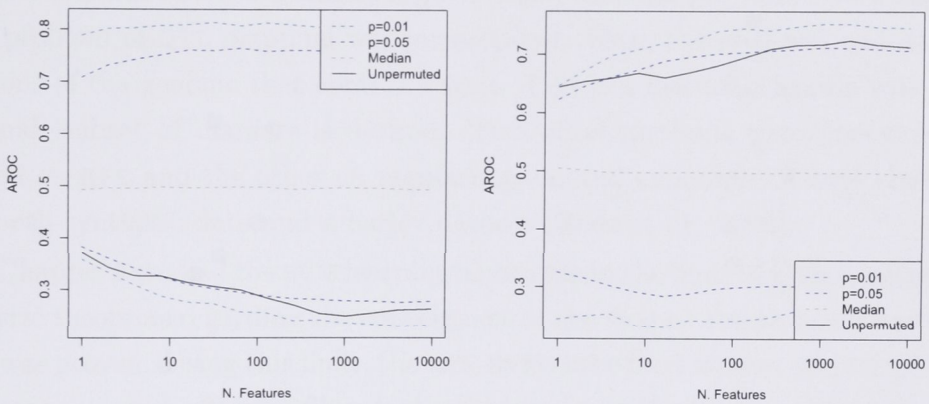


Figure 6.5: Adenocarcinoma dataset with the centroid method and the centroid method with reverser algorithms applied. Results are ϵ-0 bootstrap results with no replicates in the training set.



(a) Colon cancer dataset (b) van 't Veer dataset

Figure 6.6: Colon and van 't Veer datasets with ordinary centroid classifier and the centroid classifier with reverser algorithms applied. Results are ϵ-0 bootstrap results with no replicates in the training set. Note that all 3 plots in subfigure a are overlapping, and the cfs+centroid plot overlaps the k-Rev plot in subfigure b.



(a) Centroid (b) k-Rev centroid

Figure 6.7: Permutation test results for the centroid method and the k-Rev algorithm applied to the centroid method on the adenocarcinoma dataset. Results are ϵ-0 bootstrap results with no replicates in the training set for the k-Rev algorithm.

Chapter 7

Conclusions

This thesis has covered a large range of bioinformatics problems. In Chapter 3, the problem of QTL mapping was investigated. Here, the goal was to locate key regions of the genome that control a trait. This is a feature selection task where a small subset of markers is desired. Two novel methods were presented, one based on RFE and the other on regularisation, and experiments were conducted on both synthetic data and a barley dataset (Bedo et al., 2008).

Chapter 4 studied the SVM learning algorithm in the limit of high-regularisation. Several theorems regarding the convergence of the SVM to a simple centroid classifier was proven. Using this limit, the RFE-SVM embedded feature selection method reduces to a cost effective filter feature selection method and centroid classifier. This simple combination was evaluated on several bioinformatics datasets and shown to perform well in comparison to the RFE-SVM and other methods. One possible explanation for this performance is that bioinformatics datasets require a large amount of regularisation as they have few samples. As the centroid is the high-regularisation limit of support vector machines, this would explain its good performance despite the simplicity of the classifier.

Section 5.1 turns to the problem of unsupervised feature selection under the inspiration of a microarray design problem. The problem studied in this chapter was to design a new microarray plate for sugarcane by selecting a small subset of clones (approximately 7,000) out of a pool of 50,000. The selection of clones must remain *generic* for prediction of any trait and not targeted towards specific traits. To this end, an unsupervised feature selection method based on the Hilbert–Schmidt independence criterion and quantum annealing was proposed and evaluated. The results were promising with several bioinformatics datasets able to be compressed without deterioration of the overall predictive performance.

In addition to the unsupervised selection algorithm, an alternative derivation of the HSIC bypassing explicit use of Hilbert–Schmidt operators was presented, along with a rigourous proof for an estimator of the HSIC.

Finally, Chapter 6 studied the phenomenon of anti-learning. Anti-learning is characterised by consistent misclassification of withheld test samples despite high training accuracy. This is distinct to the problem of overfitting as the level of performance is significantly below random guessing. The presence of antilearning in natural data (an adenocarcinoma dataset) was shown to be significant, and a method for detecting anti-learnable signatures and correcting the classifier output is presented.

7.1 Summary of Contributions

Though the underlying theme has been feature selection, each chapter has focused on individual problems and proposed significantly different solutions. In Chapter 3, two methods of QTL mapping were proposed and evaluated. The RFE based method has been published (Bedo et al., 2008), but the second method remains unpublished. Both methods are novel and considerably different to the methods currently used in the area. They have been used commercially by DArT Pty. Ltd. for QTL analysis for crop breeders.

The centroid chapter (Chapter 4) has two main contributions: the theoretical link between the SVM and a centroid classifier in the high-regularisation limit, and the centroid feature filter and classifier combination. The centroid feature filter and classifier combination was shown to perform well when compared to various other learning methods. A paper describing an initial link between high-regularisation SVM and the centroid classifier has been published (Bedo et al., 2006), but the treatment here has improved proofs and additional results regarding the convergence of performance metrics measured on SVM in the high limit. The latter is derived from a paper published by Kowalczyk (2007b).

In Section 5.1, a method of unsupervised feature selection, UBHSIC, was presented and evaluated. The UBHSIC method is an extension of prior joint work on supervised selection using the HSIC (Song et al., 2007b,a), and a portion of the work presented in this chapter has been published (Bedo, 2008). The proposed quantum annealing is novel as Song et al. (2007b,a) made use of backward elimination and is currently unpublished. The application of microarray design for sugarcane crops is novel and currently unpublished. UBHSIC has direct potential

for commercial application to manufacturing specialised microarrays.

In addition to the unsupervised selection algorithm, Section 5.1 presented an alternative derivation of the HSIC through the framework of MMD. This MMD framework is much simpler than the original derivation using Hilbert–Schmidt operators. A new theorem presenting an estimator of the HSIC were proven (Theorem 5.5); this theorem is the analogue of Theorem 1 by Gretton et al. (2005), but is more rigourous and addresses several flaws in the original proof.

Finally, Chapter 6 presents some published work (Kowalczyk et al., 2007) on the existence of antilearning in the adenocarcinoma dataset, but also presents new experiments on synthetic data and a novel method for detecting and reversing the behaviour in the presence of anti-learnable data.

7.2 Future Work

The QTL analysis methods presented in Chapter 3 are simplistic models that ignore many complicating factors. In particular, they do not handle any genomic structure evident within sub-populations (population structure). This can lead to the detection of QTL that segregate the sub-populations well rather than those linked with the trait. Compensating for this population structure first requires the detection of the structure, which can be considered a form of clustering. For this purpose, the HSIC could be used.

The theorems in Chapter 4 for the convergence of performance metrics are only for specific classes of maps (real analytic maps and C^k maps to finite space with a polynomial perturbation). It may be possible to extend the theorems to include C^k maps to infinite space, but the way to proceed is unclear.

The unsupervised feature selection algorithm, UBHSIC, was proposed in conjunction with a quantum annealing optimisation method to find a solution. Though the results were good, a better method of optimisation may be available. If tight upper and lower bounds could be derived, then a branch and bound approach may be a suitable alternative. Alternatively, the problem could be cast into an unconstrained sparsity recovery problem by introducing a sparse L^1 regulariser over the state vector and using feature weighting. Finally, the UBHSIC method scales with $O(n^2)$ where n is the number of samples. While this is currently a minor problem due to the prevalence of small sample sizes in bioinformatics, it does prevent the application of the technique to large sample sizes when they are available. Further strategies of applying UBHSIC to large sample sizes may be

developed, for example by selection of small sets of key *support vectors*.

The UBHSIC method presented in Chapter 5 can be extended to perform simultaneous clustering and feature selection. Here, the goal is to select a few features that group the samples together well into a set of finite labels. The optimum set of labels and features can be sought by maximising the dependence of the reduced dataset and a kernel defined over the labels using the HSIC and the quantum optimiser. This is a form of *biclustering*.

Finally, the area of anti-learning currently remains mostly unexplored. The underlying processes that cause anti-learnable structures are not well understood. Further synthetic models and experiments need to be conducted to understand the phenomenon in more detail. Also note that the sufficient condition used to generate the synthetic antilearnable data is not a necessary condition; the derivation of *sufficient and necessary* conditions for antilearning would further progress the area considerably.

The reversible methods presented in Chapter 6 predict normally on the independent test data, but consistently misclassify the training data. Arguably the generalisation ability is of primary concern as the sample size is very small, however the real-life benefits need to be ascertained by further studies on natural data. Consistent classifiers that correctly classify the training data and generalises are desirable. Such consistent classifiers are possible, however it is unclear how to induce them.

7.3 Concluding Remarks

The field of bioinformatics is extremely diverse, and this thesis stands as a testament to this. Each chapter has studied different problems, united by the use of feature selection and microarray data, and yet they remain as very distinct problems. This large diversity provides many opportunities for development of machine learning techniques. Furthermore, with the current increase in the resolution of technologies, e.g., the recent 1.8 million SNP chips being produced by Affymetrix, the ratio of sample size to the number of features is decreasing rather than increasing, requiring development of machine learning techniques able to learn on very few samples of data. These small sample sizes can give rise to phenomena such as the anti-learning concept presented in Chapter 6, and no doubt more will be encountered in the future. In short, bioinformatics provides many opportunities and surprises, and is an excellent area for practical and theoretical

statistical machine learning.

Appendix A

Genome Profiles for Barley Data

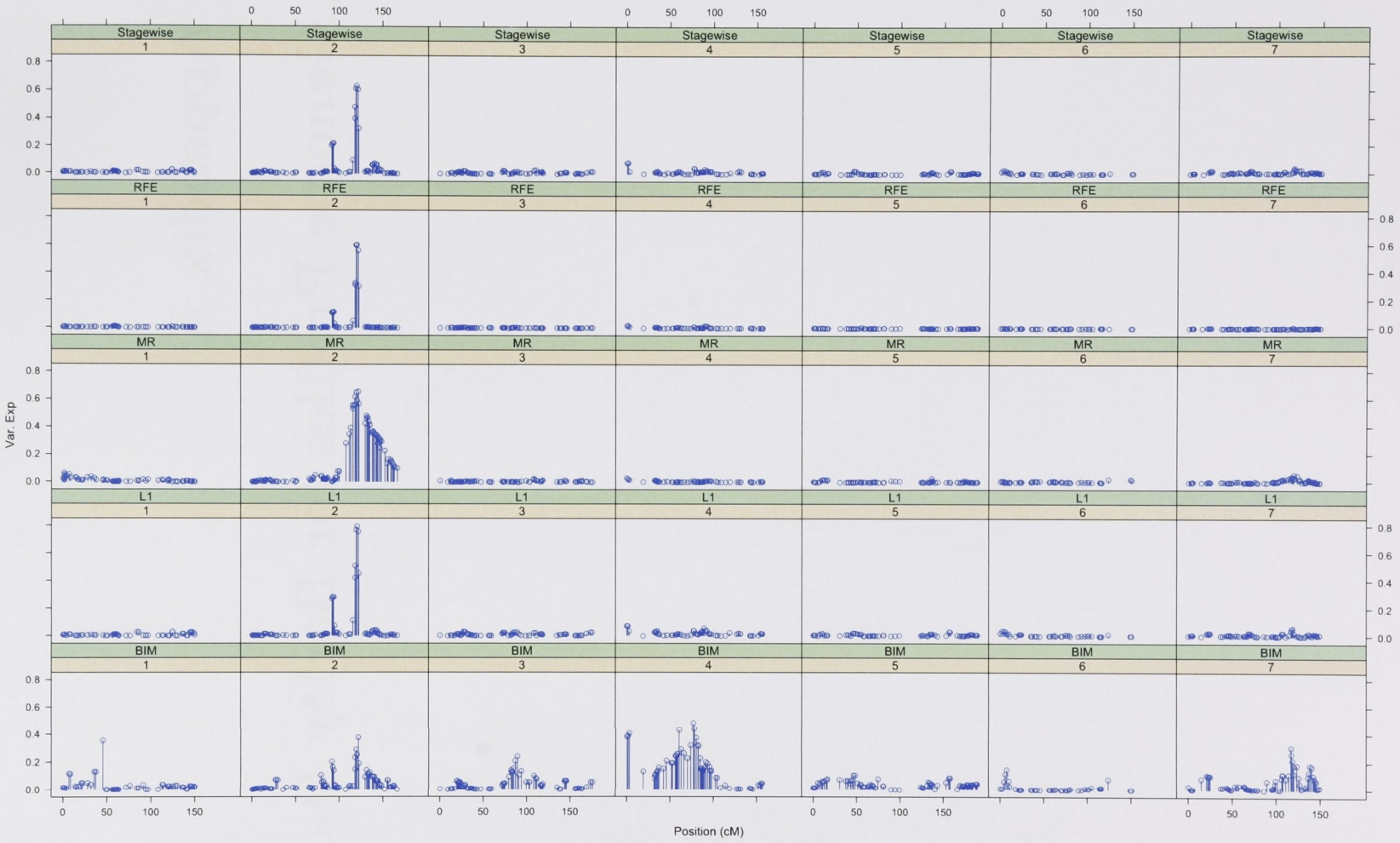


Figure A.1: Results for the days to heading phenotype on DArT natural data. Genome profiles generated using several different methods: Stagewise, RFE, L^1, and BIM.

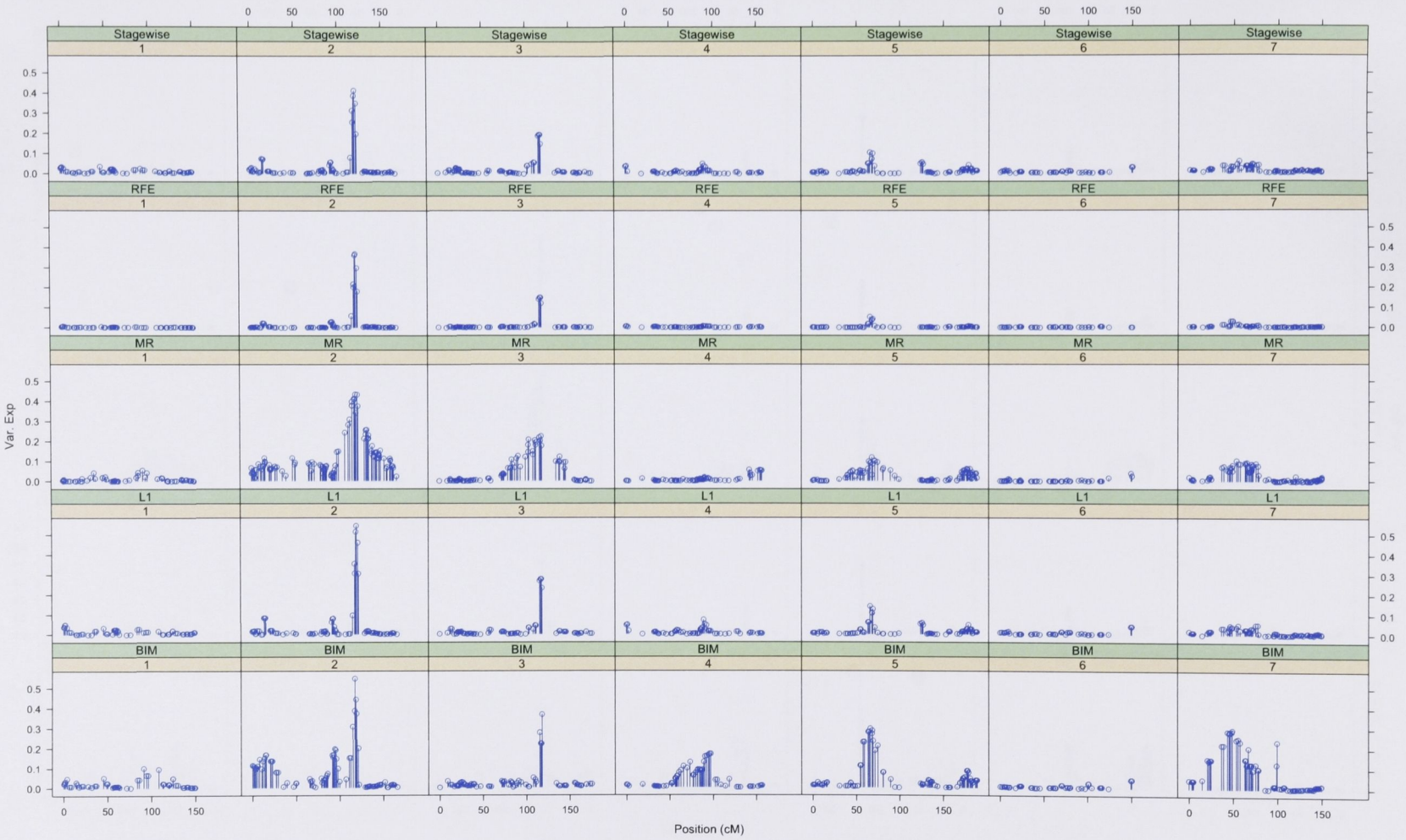


Figure A.2: Results for the height phenotype on DArT natural data. Details are as Figure A.1.

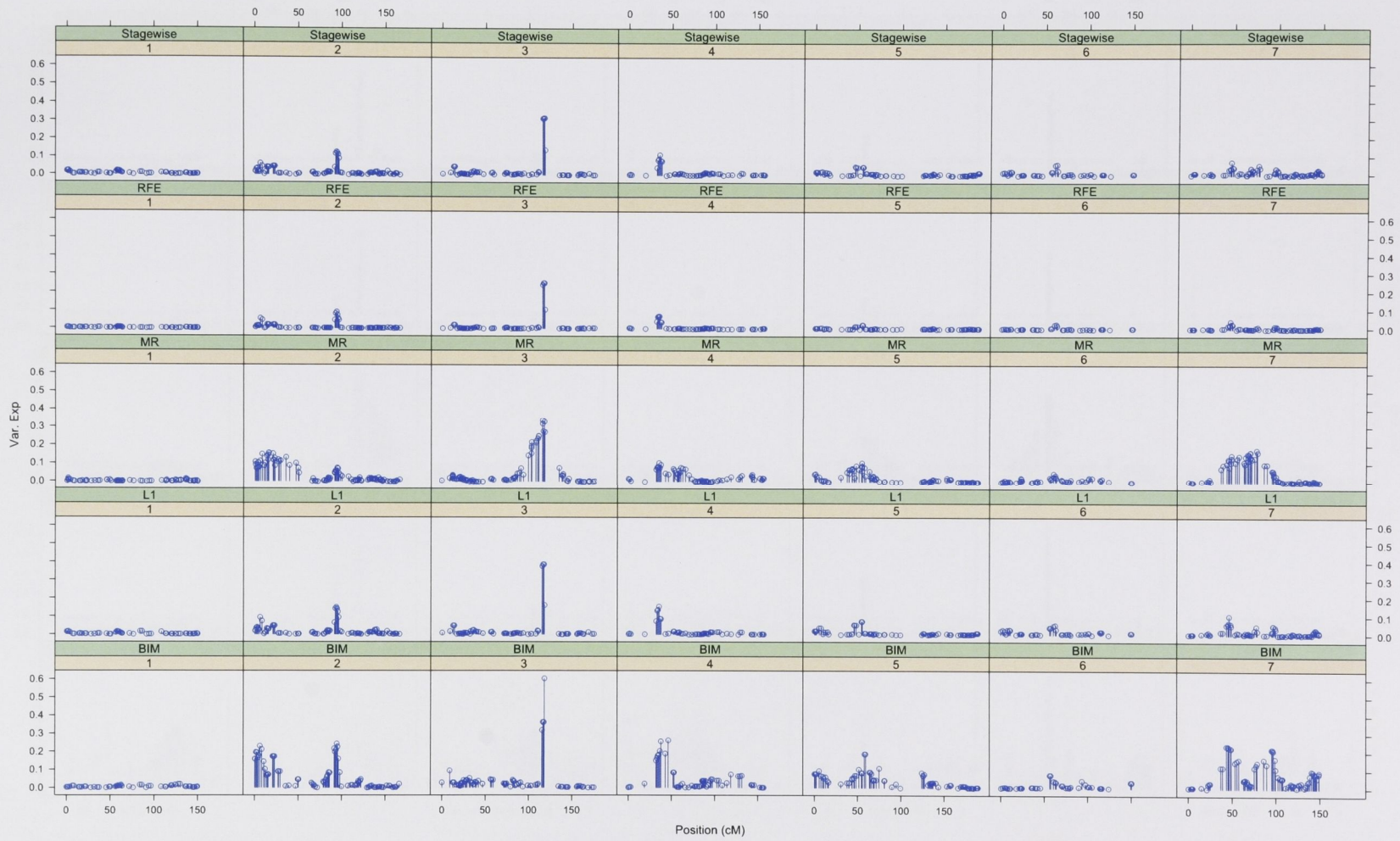


Figure A.3: Results for the lodging phenotype on DArT natural data. Details are as Figure A.1.

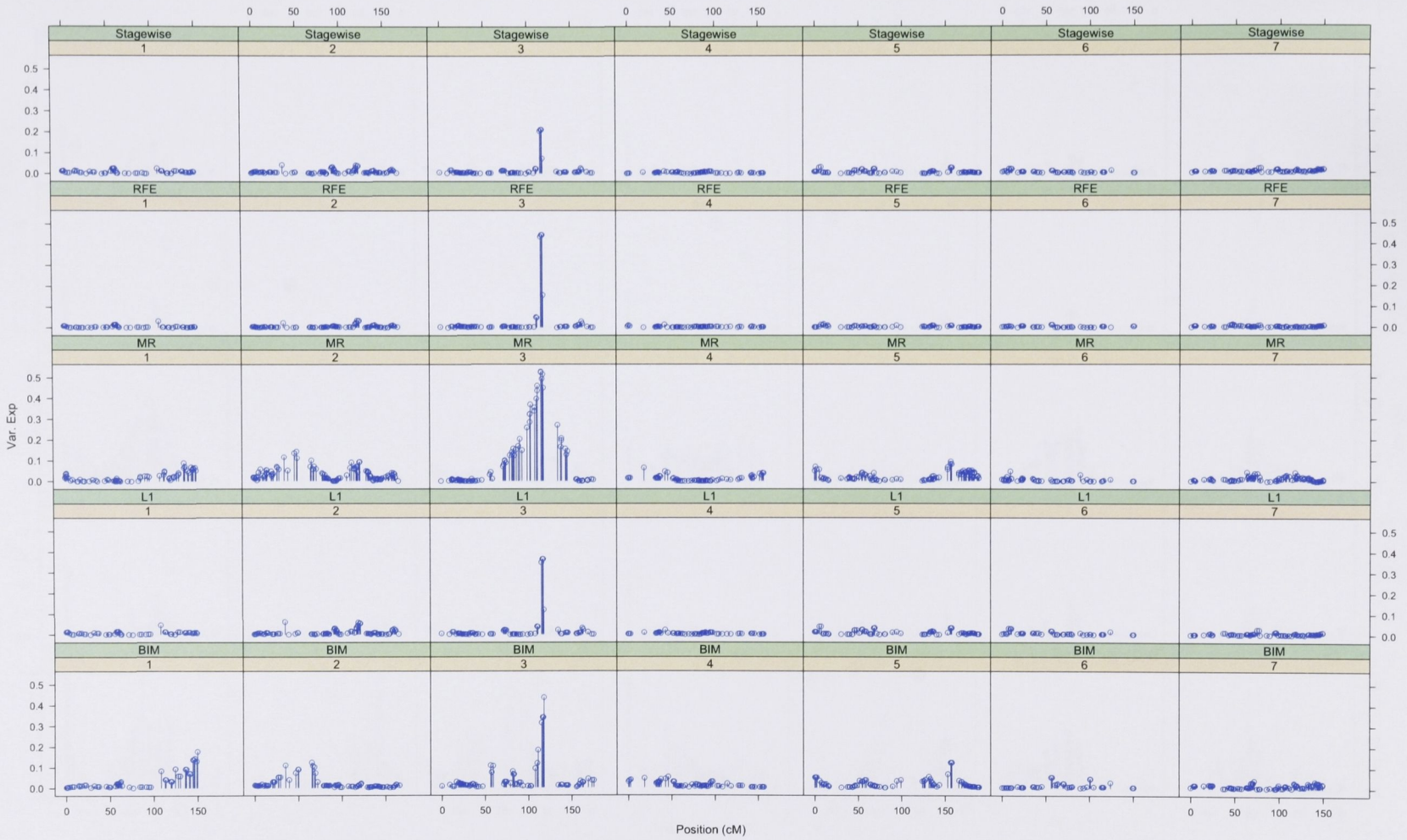


Figure A.4: Results for the yield phenotype on DArT natural data. Details are as Figure A.1.

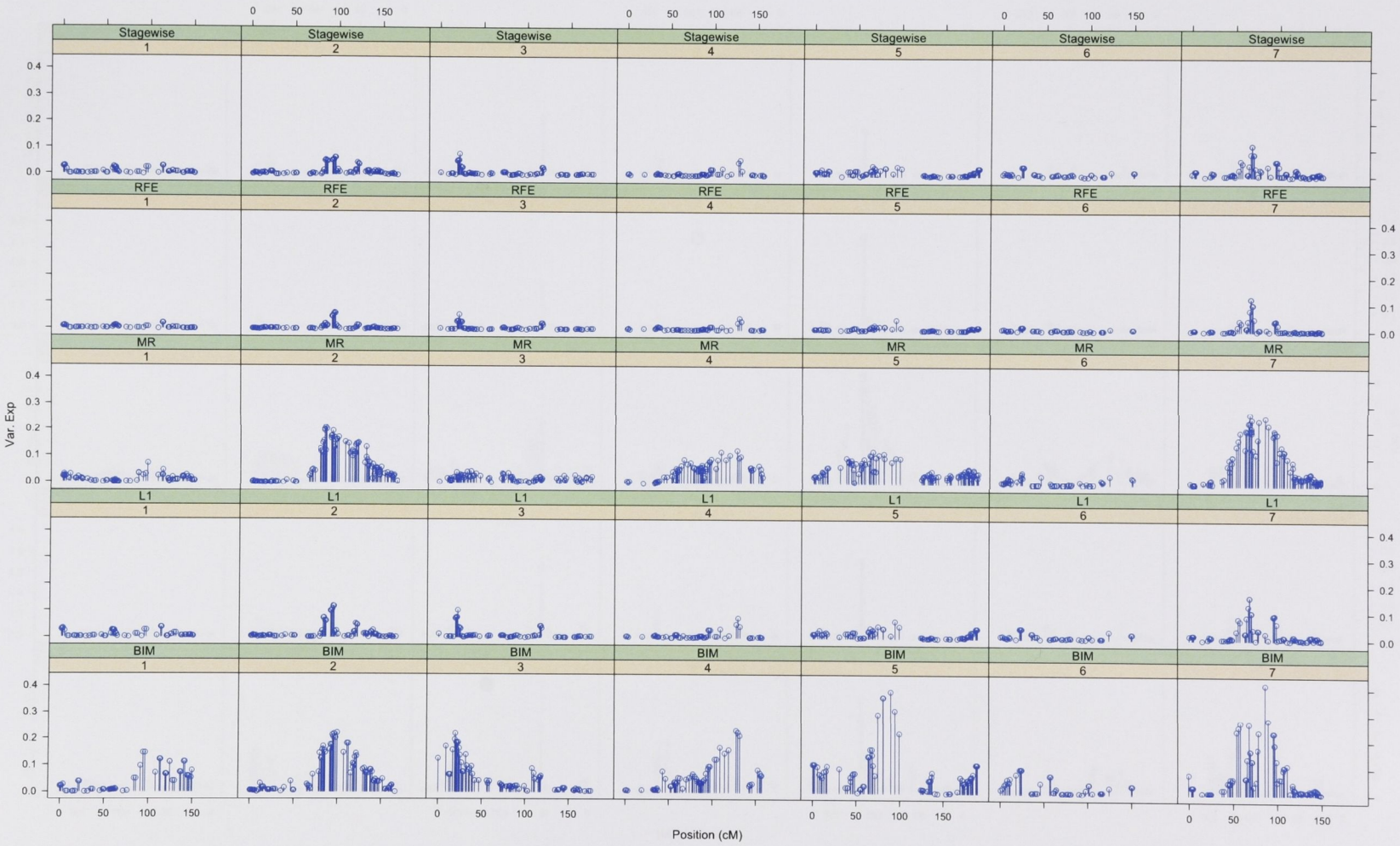


Figure A.5: Results for the α-amylase phenotype on DArT natural data. Details are as Figure A.1.

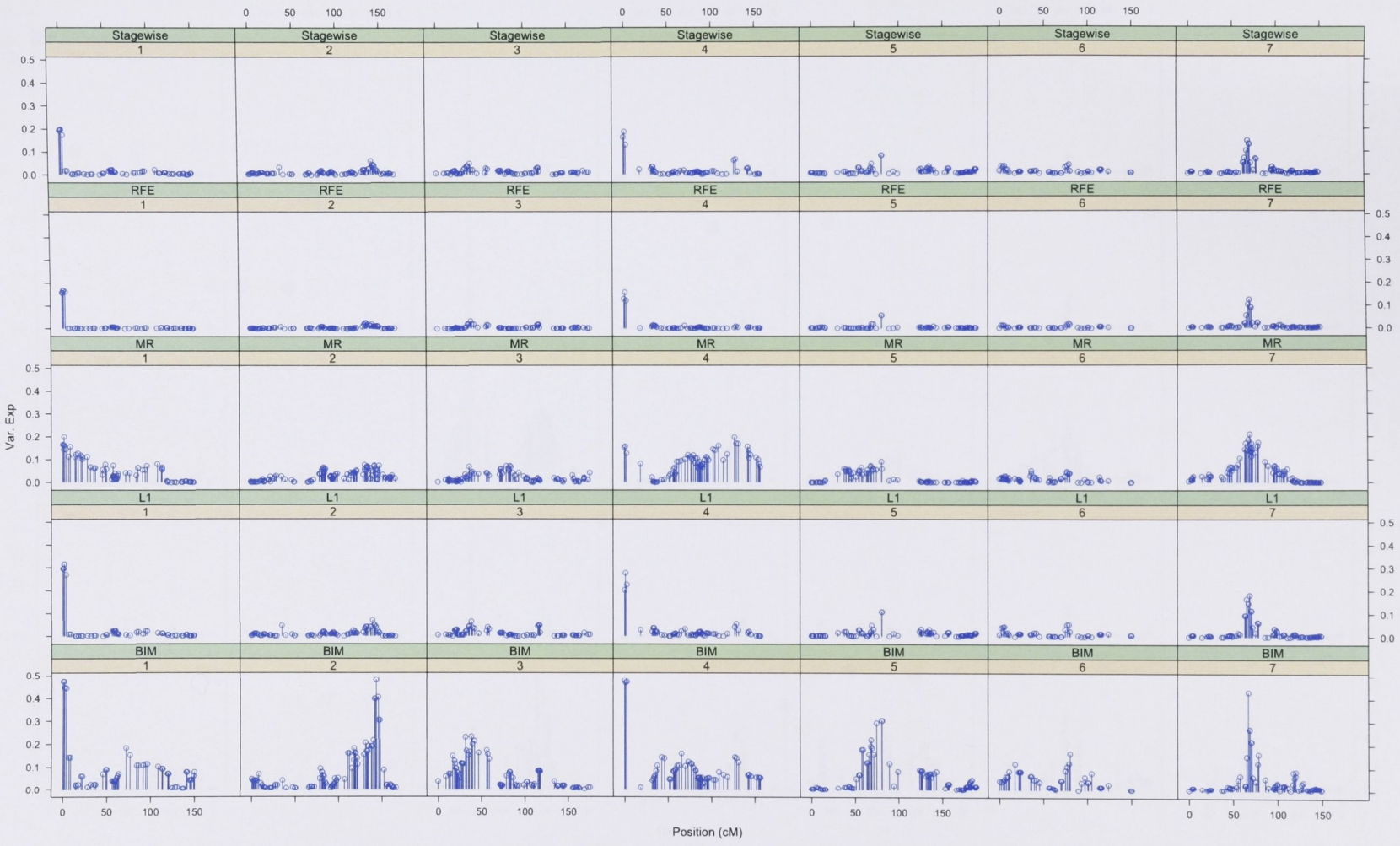


Figure A.6: Results for the diastatic power phenotype on DArT natural data. Details are as Figure A.1.

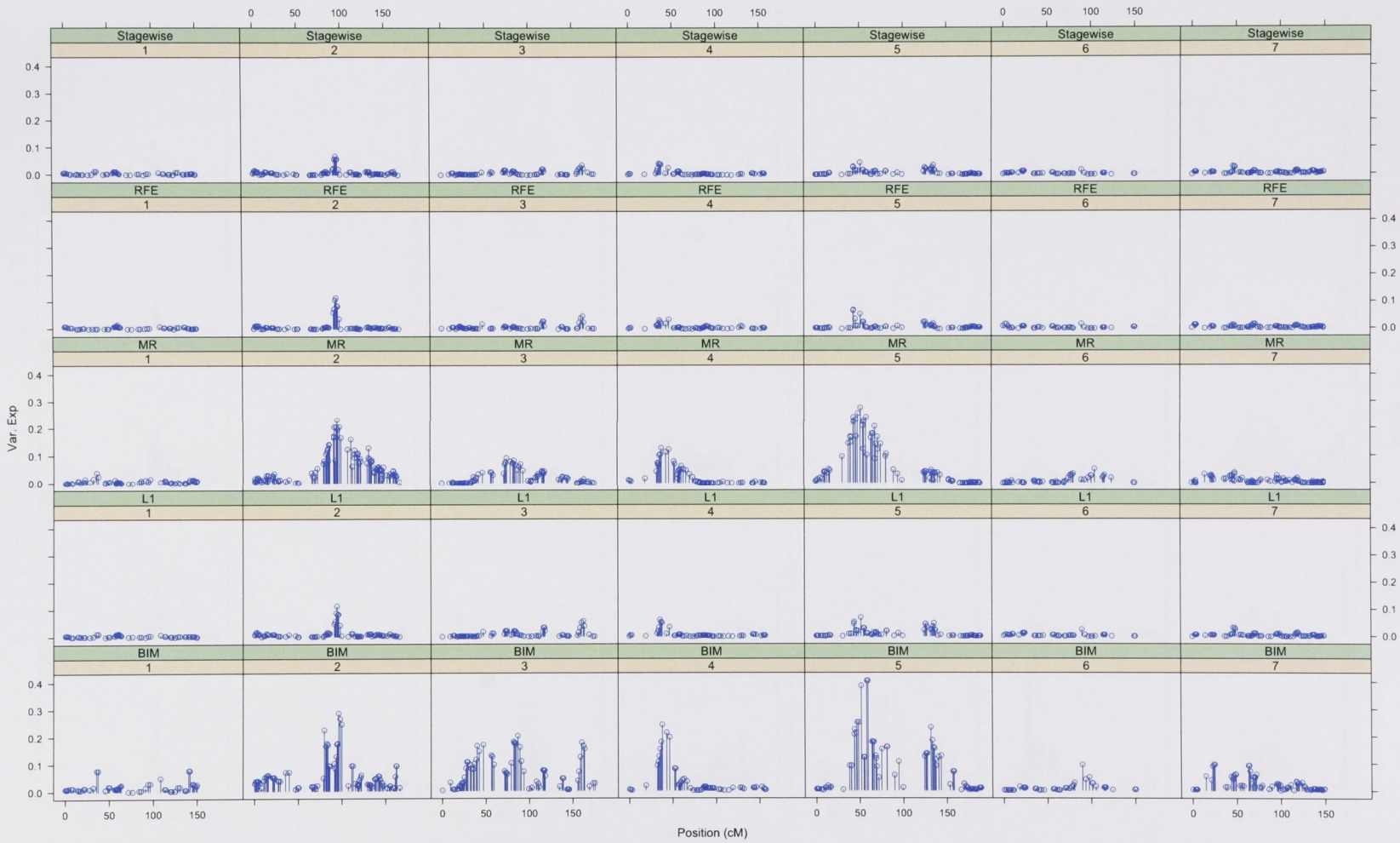


Figure A.7: Results for the protein content phenotype on DArT natural data. Details are as Figure A.1.

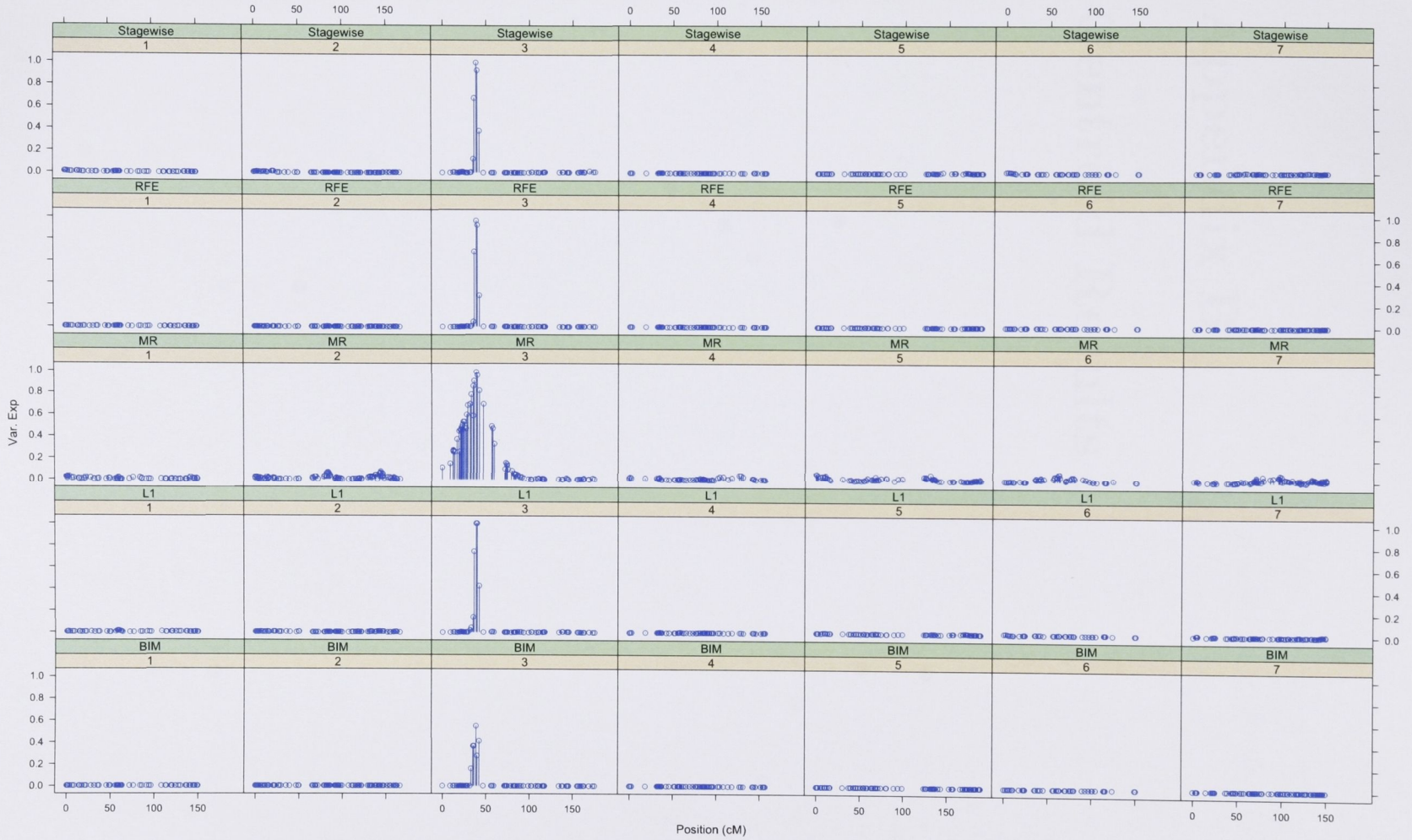


Figure A.8: Results for the pubescence phenotype on DArT natural data. Details are as Figure A.1.

Appendix B

Centroid Results

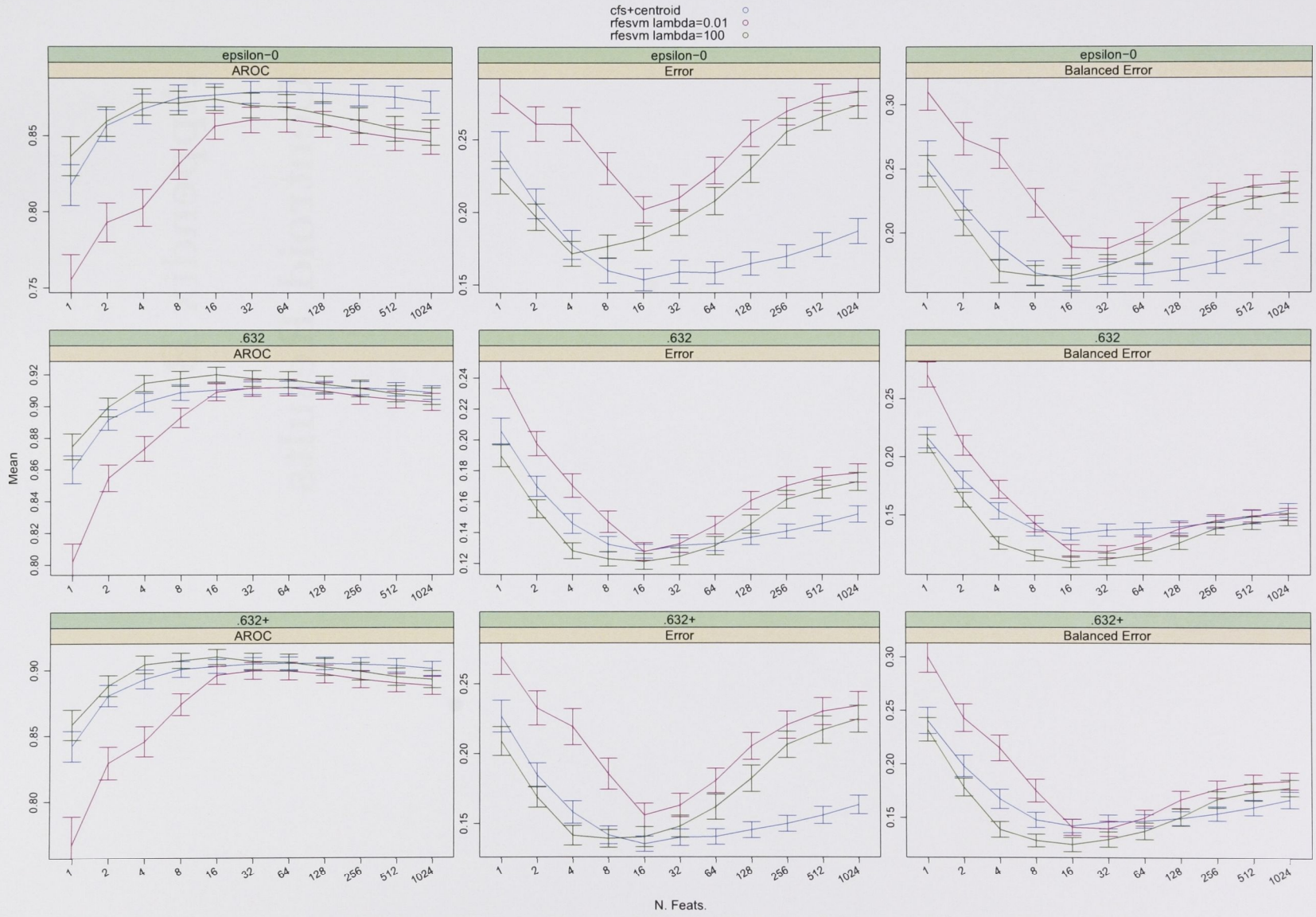


Figure B.1: Bootstrap results for centroid and RFE–SVM on *colon* dataset. Error bars are 95% confidence intervals.

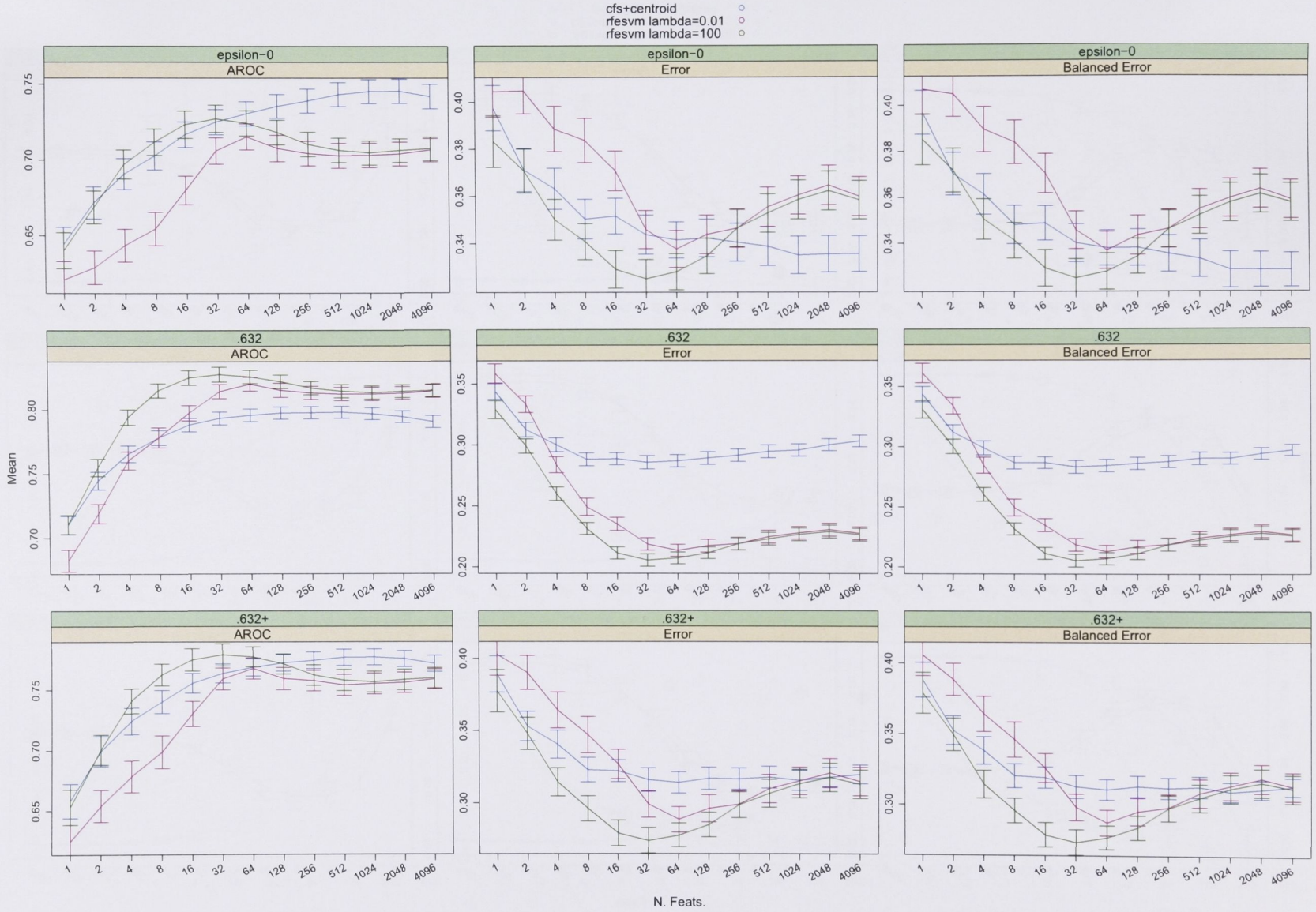


Figure B.2: Bootstrap results for centroid and RFE–SVM on *Van 't Veer* dataset. Error bars are 95% confidence intervals.

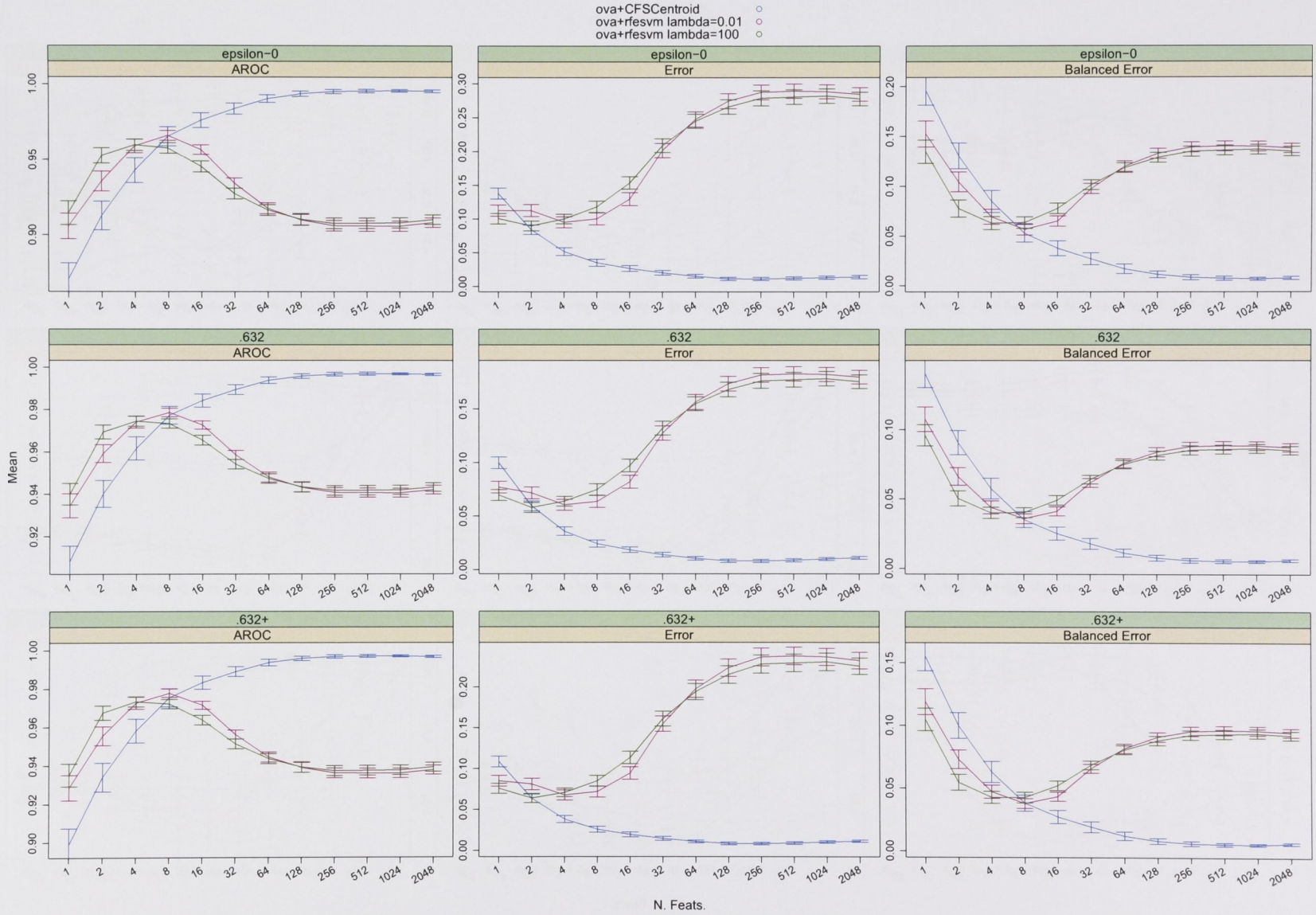


Figure B.3: Bootstrap results for centroid and RFE–SVM on *lymphoma* dataset. Error bars are 95% confidence intervals.

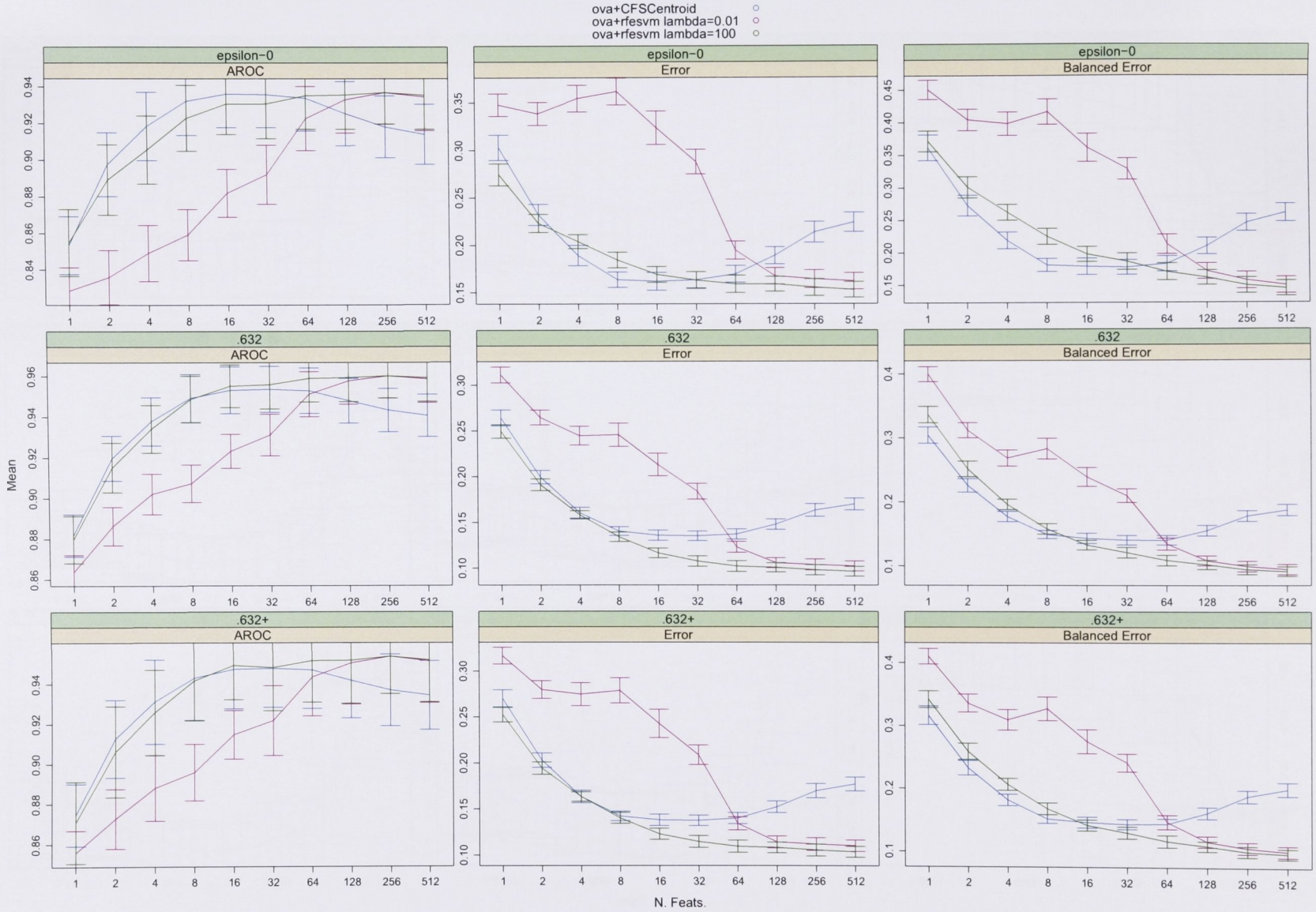


Figure B.4: Bootstrap results for centroid and RFE–SVM on *CUP* dataset. Error bars are 95% confidence intervals.

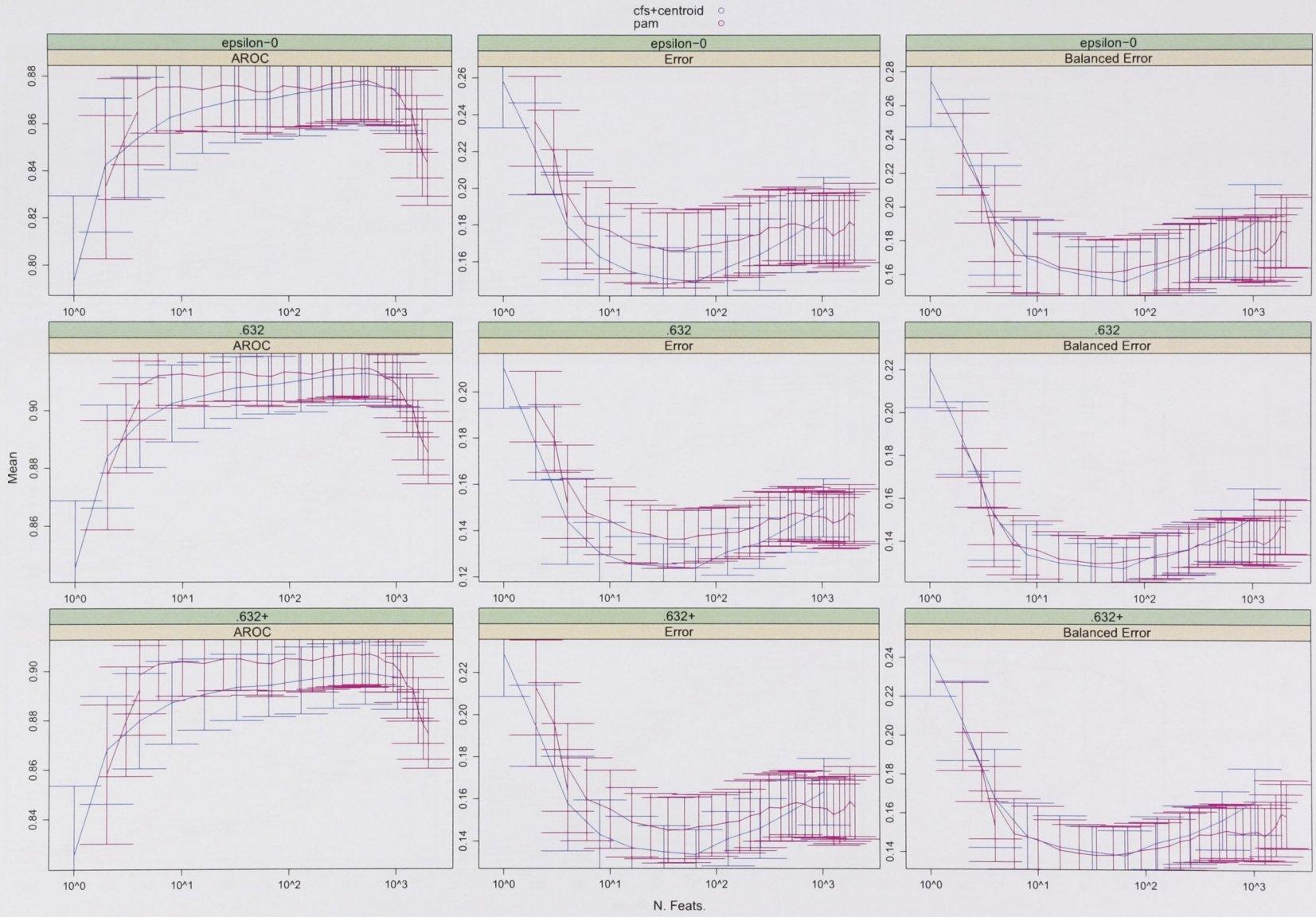


Figure B.5: Bootstrap results for centroid and PAM on *colon* dataset. Error bars are 95% confidence intervals.

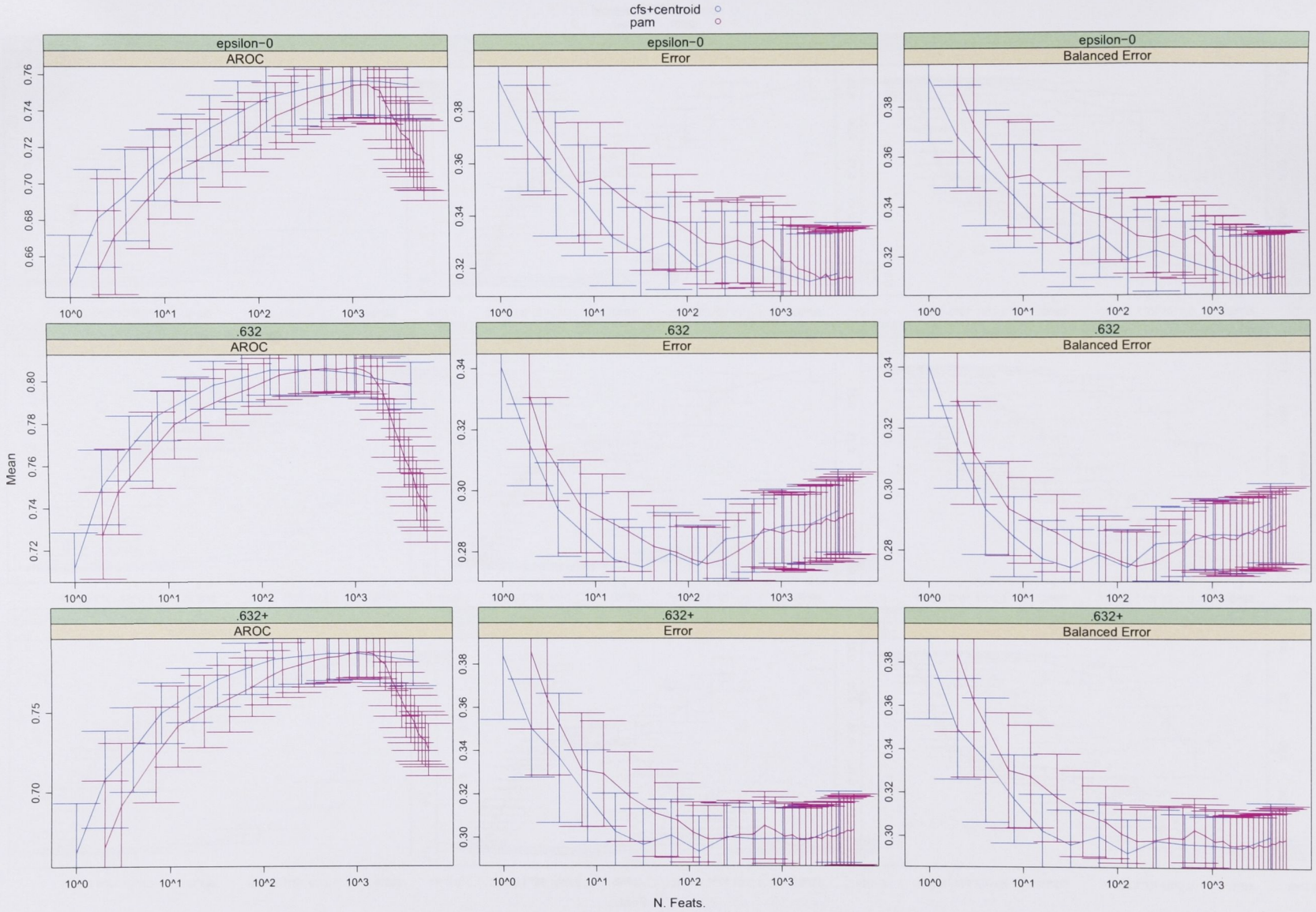


Figure B.6: Bootstrap results for centroid and PAM on *Van 't Veer* dataset. Error bars are 95% confidence intervals.

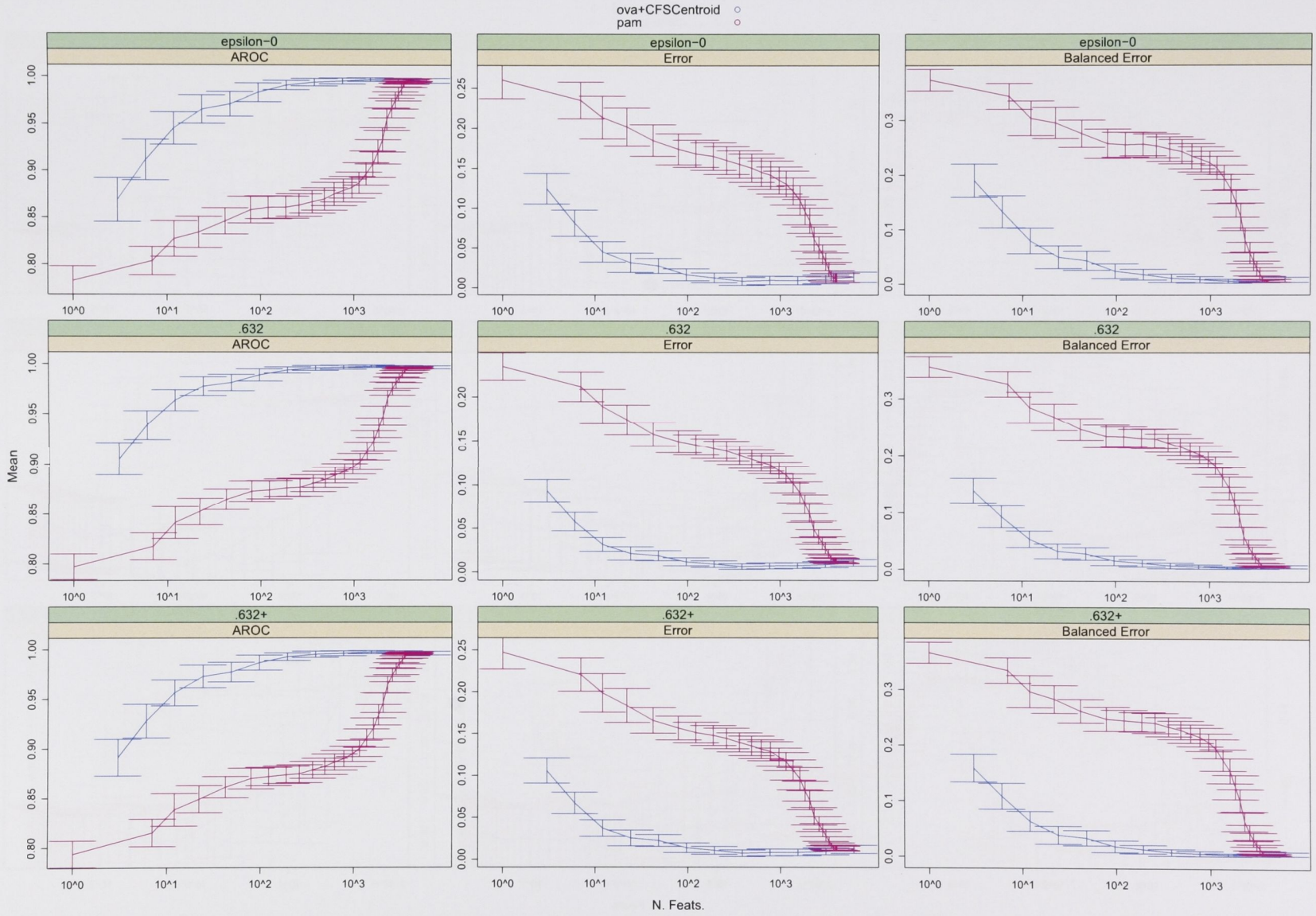


Figure B.7: Bootstrap results for centroid and PAM on *lymphoma* dataset. Error bars are 95% confidence intervals.

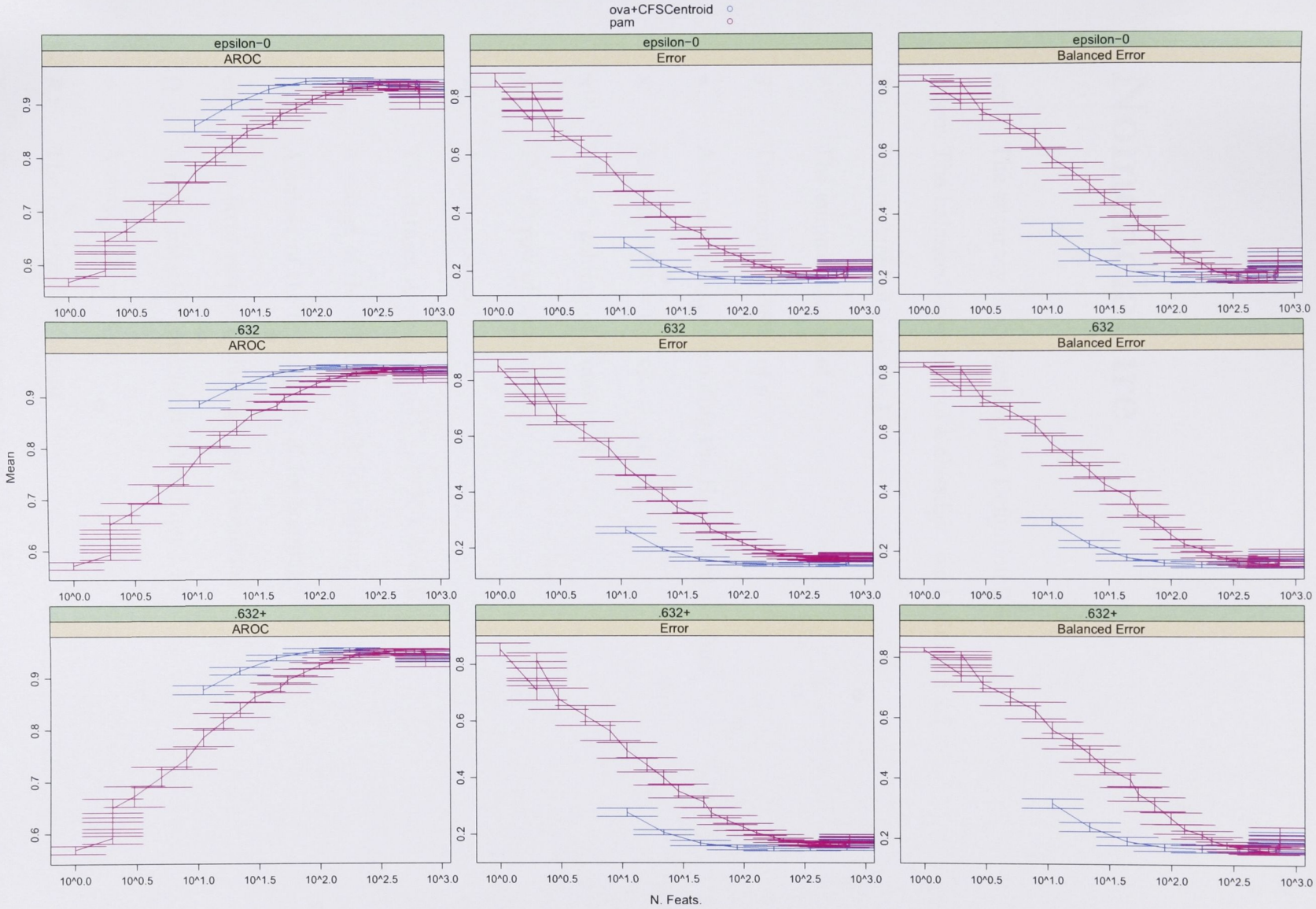


Figure B.8: Bootstrap results for centroid and PAM on *CUP* dataset. Error bars are 95% confidence intervals.

Nomenclature

acc	The accuracy of a classifier, see Definition 2.6
balerr	The balanced error rate of a classifier, see Definition 2.7
$\bar{\mathbf{x}}$	The centroid of a set $\{\mathbf{x}_i\}$
$\bar{x}$	The empirical mean of x
$\bar{y}$	The empirical mean of y
$\bullet$	A binary operator denoting the Hadamard product
$\mathbf{x}$	An n-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_n)$
$\mathbf{x}^{(j)}$	The j^{th} column of a matrix X
$\mathbf{x}_i$	The i^{th} row of a matrix X
$\mathcal{H}$	A Hilbert space
$\mathcal{X}$	The dataset $\mathcal{X} := \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \subset \mathbb{X} \times \mathbb{Y} = \mathbb{R}^m \times \mathbb{R}$
$\mathcal{X}_{\text{test}}$	A subset of the full dataset $\mathcal{X}$ for testing
$\mathcal{X}_{\text{train}}$	A subset of the full dataset $\mathcal{X}$ for training
err	The error rate of a classifier, see Definition 2.6
Ω	A regularisation functional (Definition 2.19)
ϕ	A map $\phi\colon \mathbb{X} \to \mathcal{H}$
$\mathbb{R}$	The real number system
$\mathbb{X}$	The set a dataset is drawn from ($\mathbb{X} \subset \mathbb{R}^m$)

$\mathbb{Y}$ The set of possible target values: for regression $\mathbb{Y} = \mathbb{R}$, for binary classification $\mathbb{Y} = \{1, -1\}$, and for multiclass classification $\mathbb{Y}$ is a finite set

Id The identity matrix

K A *kernel matrix* with elements $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ where k is the kernel function

$L\colon \mathbb{Y} \times \mathbb{R} \to [0, \infty)$ Loss function (Definition 2.15)

m The number of features

n The number of samples

r^2 Proportion of variance explained, see Definition 2.5

T_2 A Hausdorff space (see Definition 4.3)

AIC Akaike information criterion (Definition 2.31)

AROC Area under the ROC (Definition 2.9)

BC Backcross experiment (Figure 3.1)

BIC Bayesian information criterion (Definition 2.32)

BIM Bayesian interval mapping, see Section 3.1.7

CFS Centroid feature selection, see Section 4.6

CFSCentroid CFS coupled with the centroid classifier, see Section 4.6

DMC Diffusion Monte Carlo, see Section 5.2

DNA Deoxyribonucleic acid

DSC Direct sufficient condition method for generating synthetic anti-learnable data. See Chapter 6.

EM Expectation Maximisation (Dempster et al., 1977)

FPR The False Positive Rate. See Definition 2.8.

GOF Goodness of fit, see Section 2.2

HSIC Hilbert–Schmidt independence criterion

IID Independent and Identically Distributed.

KL-divergence Kullback-Leibler divergence (Definition 2.25)

KNN k-nearest neighbours classifier

LIMMA Linear Models for Microarray Analysis. See Section 2.5.

LOD Logarithm of Difference (Section 3.1.2)

LOO Leave one out, see Section 2.6.2

LTO Leave two out, see Section 2.6.2

MC Monte Carlo

MCMC Markov chain Monte Carlo

MMD Maximum mean discrepancy, see Chapter 5

MR Marker regression, see Section 3.1.4

OFP Orthogonal frame projection method for generating synthetic anti-learnable data. See Chapter 6.

OVA One-vs-all multiclass architecture (Section 2.8)

PAM The shrunken centroid classifier, see Section 4.7.2

PAVE Predictive apportioned variance explained

QA Quantum annealing

QPCR Quantitative real time polymerase chain reaction

QTL Quantitative Trait Locus/Loci (Section 3.1)

RBF Radial basis function kernel, see Example 2.14

RFE Recursive Feature Elimination. See Section 2.5.

RFE–SVM Recursive feature elimination support vector machine

RKHS A Reproducing Kernel Hilbert Space

ROC Receiver Operating Characteristic (Figure 2.1)

RSS Residual Sum of Squares, see Definition 2.4

SAM Significance of Microarrays. See Section 2.5.

SCC Squamous cell carcinoma

SML Statistical Machine Learning

SNP Single nucleotide polymorphism

SNR Signal to noise ratio (Section 2.5)

SVM Support Vector Machine, see Proposition 2.22 and Proposition 2.24

TPR The True Positive Rate. See Definition 2.8.

UBHSIC Unsupervised feature selection By the Hilbert–Schmidt independence criterion, see Chapter 5

cM centiMorgans, a measure of genetic distance

Hinge loss Equation 2.4

M Morgans, a measure of genetic distance

Quadratic loss Equation 2.4

Sensitivity See TPR.

Specificity $1 -$ FPR. See FPR.

AUC See AROC

Bibliography

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

Albert, A. (1972). *Regression and the Moore-Penrose Pseudoinverse*. Academic Press.

Alberts, B., Johnsno, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell*. Garland, 4th edition.

Aldous, D. and Vazirani, U. (1994). "go with the winners" algorithms. *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 492–501.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750.

Ambroise, C. and McLachlan, G. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences*, 99(10):6562.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404.

Balding, D. J., Bishop, M., and Cannings, C., editors (2001). *Handbook of Statistical Genetics*. John Wiley & Sons.

Bedo, J. (2008). Microarray design using the hilbert—schmidt independence criterion. *Proceedings of the Third IAPR International Conference on Pattern Recognition in Bioinformatics*, pages 288–298.

Bedo, J., Sanderson, C., and Kowalczyk, A. (2006). An efficient alternative to svm based recursive feature elimination with applications in natural language processing and bioinformatics. *Proceedings of the Australian Joint Conference on Artifical Intelligence.*

Bedo, J., Wenzl, P., Kowalczyk, A., and Kilian, A. (2008). Precision-mapping and statistical validation of quantitative trait loci by machine learning. *BMC Genet*, 9(1):35.

Belisle, C. (1992). Convergence theorems for a class of simulated annealing algorithms on r d. *Journal of Applied Probability*, 29.

Berlinet, A. and Thomas-Agnan, C. (2003). *Reproducing Kernel Hilbert Spaces in Probability and Statistics.* Springer.

Bishop, C. and Tipping, M. (2003). Bayesian regression and classification. *Advances in Learning Theory: Methods, Models and Applications*, 190:267–285.

Boser, B., Guyon, I., and Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational Learning Theory*, pages 144–152.

Braga-Neto, U. and Dougherty, E. (2004). Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–80.

Broman, K. (2001). Review of statistical methods for qtl mapping in experimental crosses. *Lab animal*, 30(7):44–52.

Broman, K. (2002). A model selection approach for the identification of quantitative trait loci in experimental crosses. *J. R. Statist. Soc. B*, 64:641–656.

Broman, K., Wu, H., Sen, S., and Churchill, G. (2003). R/qtl: Qtl mapping in experimental crosses. *Bioinformatics*, 19(7):889–890.

Candès, E. and Tao, T. (2005). Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215.

Chen, S., Donoho, D., and Saunders, M. (1998). Atomic decomposition by basis persuit. *SIAM J. Sci. Comput.*, 20:33–61.

Churchill, G. and Doerge, R. (1994). Empirical threshold values for quantitative trait mapping. *Genetics*, 138(3):963–71.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Demazure, M. (2000). *Bifurcations and Catastrophes: Geometry of Solutions to Nonlinear Problems*. Springer.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Doerge, R., Zeng, Z. B., and Weir, B. (1997). Statistical issues in the search for genes affecting quantitative traits in experimental populations. *Statistical Science*, 12(3):195–219.

Donoho, D., Elad, M., and Temlyakov, V. (2005). Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–17.

Dudley, R. (1987). *Real Analysis and Probability*. Cambridge University Press.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association*, 81(394):461–470.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, pages 407–451.

Efron, B. and Tibshirani, R. (1994). *An Introduction to the Bootstrap*. Chapman & Hall/CRC.

Efron, B. and Tibshirani, R. (1997). Improvements on cross-validation: The. 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560.

Fortet, R. and Mourier, E. (1953). Convergence de la répartition empirique vers la répartition théorique. *Annales scientifiques de l'École Normale Supérieure Sér. 3.*

Freund, Y. and Schapire, R. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296.

Gabriel, K. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, 58(3):453–467.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC, 2nd edition.

Gerstein, M., Bruce, C., Rozowsky, J. S., Zheng, D., Du, J., Korbel, J. O., Emanuelsson, O., Zhang, Z. D., Weissman, S., and Snyder, M. (2007). What is a gene, post-encode? history and updated definition. *Genome Res*, 17(6):669–81.

Golub, T., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E. S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–7.

Golubitsky, M. and Guillemin, V. (1974). *Stable Mappings and Their Singularities*. Springer.

Greenawalt, D. M., Kowalczyk, A., and Ciavarella, M. (2007). Pretreatment gene expression profiles can be used to predict response to neoadjuvant chemoradiotherapy in esophagael cancer. *Ann. Surg. Oncol.*, 14(12):3602–3609.

Gretton, A., Borgwardt, K., Rasch, M., Scholkopf, B., and Smola, A. (2006). A kernel method for the two-sample-problem. *Journal of Machine Learning Research*, 1:1–10.

Gretton, A., Bousquet, O., Smola, A., and Scholkopf, B. (2005). Measuring statistical dependence with hilbert-schmidt norms. *LECTURE NOTES IN COMPUTER SCIENCE*, 3734:63–77.

Guyon, I. (2003). An introduction to variable and feature selection. *JMLR*, 3:1157–1182.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A., editors (2006). *Feature Extraction: Foundations and Applications*. Springer.

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422.

Haley, C. and Knott, S. (1992). A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity*, 69(4):315–324.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer.

Hayes, P., Liu, B., Knapp, S., Jones, B., Blake, T., Franckoiak, J., Rasmusson, D., Sorrells, M., Ullrich, S., Wesenberg, D., and Kleinhofs, A. (1993). Quantitative trait locus effects and environmental interaction in a sample of north american barley germ plasm. *TAG*, 87:392–401.

Hocking, J. G. and Young, G. S. (1988). *Topology*. Dover Publications.

Huang, T. and Kecman, V. (2005). Gene extraction for cancer diagnosis by support vector machines—an improvement. *Artificial Intelligence In Medicine*, 35:185–194.

Jansen, R. C. (1993). Interval mapping of multiple quantitative trait loci. *Genetics*, 135(1):205–11.

Kao, C.-H., Zeng, Z. B., and Teasdale, R. D. (1999). Multiple interval mapping for quantitative trait loci. *Genetics*, 152(3):1203–16.

Kearsey, M. and Hyne, V. (1994). Qtl analysis: a simple 'marker-regression'approach. *TAG*, 89:698–702.

Knapp, S. (1991). Using molecular markers to map multiple quantitative trait loci: models for backcross, recombinant inbred, and doubled haploid progeny. *TAG*, 81:333–338.

Knapp, S., Bridges, W., and Birkes, D. (1990). Mapping quantitative trait loci using molecular marker linkage maps. *TAG*, 79(583-592).

Kosztin, I., Faber, B., and Schulten, K. (1997). Introduction to the diffusion monte carlo method. *Arxiv preprint physics/9702023*.

Kowalczyk, A. (2007a). Classification of anti-learnable biological and synthetic data. *KDD*, 4702:176.

Kowalczyk, A. (2007b). Continuity of performance metrics for thin feature maps. *ALT*.

Kowalczyk, A., Greenawalt, D. M., Bedo, J., Cuong, D., Raskutti, G., Thomas, R. J., and Phillips, W. A. (2007). Large validation of anti-learnable signature in classification of response to chemoradiotherapy in esophageal adenocarcinoma patients. *Lecture Notes in Operations Research*, 7:213–221.

Lander, E. S. and Botstein, D. (1989). Mapping mendelian factors underlying quantitative traits using rflp linkage maps. *Genetics*, 121(1):185–99.

Lolle, S., Victor, J., Young, J., and Pruitt, R. (2005). Plant genetics: Hothead healer and extragenomic information (reply). *Nature*.

Mann, H. and Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Annals of Mathematical Statistics*.

McLachlan, G. J., Do, K.-A., and Ambroise, C. (2004). *Analyzing Microarray Gene Expression Data*. Wiley-Interscience.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer.

Parker, B., Gunter, S., and Bedo, J. (2007). Stratification bias in low signal microarray studies. *BMC Bioinformatics*.

Rassoulzadegan, M., Grandjean, V., Gounon, P., Vincent, S., Gillot, I., and Cuzin, F. (2006). Rna-mediated non-mendelian inheritance of an epigenetic change in the mouse. *Nature*, 441(7092):469–74.

Rifkin, R. and Klautau, A. (2004). In defense of one-vs-all classification. *JMLR*.

Rockafellar, R. T. (1997). *Convex Analysis*. Princeton University.

Schölkopf, B. and Smola, A. J. (2002). *Learning with Kernels*. MIT Press.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis.* Cambridge University Press.

Smyth, G. K. (2004). Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(1):1027.

Song, L., Bedo, J., Borgwardt, K., Gretton, A., and Smola, A. (2007a). Gene selection via the bahsic family of algorithms. *Bioinformatics*, 23(13):i490.

Song, L., Smola, A., Gretton, A., Borgwardt, K., and Bedo, J. (2007b). Supervised feature selection via dependence estimation. *Proceedings of the 24th international conference on Machine Learning*, pages 823–830.

Speed, T., editor (2003). *Statistical Analysis of Gene Expression Microarray Data.* Chapman & Hall/CRC.

Steinwart, I. (2001). On the influence of the kernel on the consistency of support vector machines. *JMLR*, 2:67–93.

Steinwart, I. (2005). Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142.

Steinwart, I., Hush, D., and Scovel, C. (2006). An explicit description of the reproducing kernel hilbert spaces of gaussian rbf kernels. *IEEE Transactions on Information Theory*, 52:4635–4643.

Talagrand, M. (1987). The glivenko-cantelli problem. *Ann. Probab*, 15(3):837–870.

Tanksley, S. (1993). Mapping polygenes. *Annual Review of Genetics*, 27:205–233.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58(1):267–288.

Tibshirani, R., Hastie, T. J., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572.

Tibshirani, R., Hastie, T. J., Narasimhan, B., and Chu, G. (2003). Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, 18(1):104–117.

Tikhonov, A. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038.

Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine. *JMLR*, 1:211–244.

Tothill, R. W., Kowalczyk, A., Rischin, D., Bousioutas, A., Haviv, I., van Laar, R. K., Waring, P. M., Zalcberg, J., Ward, R., Biankin, A., Sutherland, R. L., Henshall, S. M., Fong, K., Pollack, J. R., Bowtell, D., and Holloway, A. J. (2005). An expression-based site of origin diagnostic method designed for clinical application to cancer of unknown origin. *Cancer Res*, 65(10):4031–40.

Tusher, V. G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci USA*, 98(9):5116–21.

van 't Veer, L., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–6.

Vapnik, V. (1998). *Statistical Learning Theory.* Wiley-Interscience.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory.* Springer.

Wainwright, M. (2006). Sharp thresholds for high-dimensional and noisy recovery of sparsity. *Technical Report.*

Wenzl, P., Carling, J., Kudrna, D., Jaccoud, D., Huttner, E., Kleinhofs, A., and Kilian, A. (2004). Diversity arrays technology (dart) for whole-genome profiling of barley. *Proc Natl Acad Sci USA*, 101(26):9915–20.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Willard, S. (2004). *General Topology.* Dover Publications.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2nd edition.

Xu, S. (2003). Estimating polygenic effects using markers of the entire genome. *Genetics*, 163(2):789–801.

Yi, N., Yandell, B. S., Churchill, G., and Allison, D. (2005). Bayesian model selection for genome-wide epistatic quantitative trait loci analysis. *Genetics*, 170:1333–1344.

Zeng, Z. B. (1994). Precision mapping of quantitative trait loci. *Genetics*.